

Kilobaud Microcomputing, Jan. 1979,
No. 26, p. 22-26.

See previous article p 88

0100 - 0510
J0100
J0103

OUTPUT TO CRT
Δ 04C9 TO CE 801C

"L" = SAVE TO TAPE

An Editor for 6800 BASIC Programs

Our instructor, Pete Stark, is on "sabbatical" this month, so we're presenting a couple of articles by Pete in lieu of Classroom No. 16, which will appear next month—Editors.

This is one of the most significant and useful pieces of software owners of SWTP systems will see in quite some time (especially those who do a lot of programming in BASIC).

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

Considering the difference in price between a small computer such as the SWTP 6800 system, and a large computer such as an IBM 370 or PDP-10, the small-computer BASIC systems are remarkably capable in comparison. There has been a tremendous improvement in the last year or so in the power of microcomputer BASICs, including even various formatted outputs and cas-

sette and disk routines. But there is one area in which the big-computer BASIC systems outshine the small—the facility to edit the BASIC source program easily and quickly. Here are two programs for the SWTP system that help with the editing job.

The two programs are called BASEDIT and BASLIST. There are overlapping areas between the two; they could probably have been combined into one larger program, but I chose to separate them because I could not assemble them together in my 16K system.



Pete Stark . . . hard at it.

Program 1. BASEDIT program.

```

00010          NAM    BASEDIT
00020          *BASIC EDITOR
00030          *P. STARK 2/78
00040 0100      ORG    $0100
00050          OPT    0
00060          *ROM ENTRY POINTS
00070          E1AC  INEEE EQU  $E1AC
00080          E1D1  OUTEEE EQU  $E1D1
00090          E0E3  CONTRL EQU $E0E3
00100          *HARD/SOFT STARTS
00110 0100 7E 0106 HARDST JMP  START
00120 0103 7E 012A SOFTST JMP  ASK
00130 0105 CE 0103 START  LDX  #$0103
00140 0109 FF A043 START  STX  $A04B
00150 010C 9E A042 START  LDS  #$A042
00160          *START WITH LF
00170 010F 86 0A  LDA A  #$0A
00180 0111 B7 0498  STA A  TEXT
00190 0114 CE 0499  LDX  #TEXT+1
00200 0117 3D 43  BSR  CRLF
00210 0119 8D E1AC READ  JSR  INEEE
00220 011C A7 00  STA A  0,X
00230          *EXIT ON CTRL E
00240 011E 31 05  CMP A  #$05
00250 0120 27 08  BEQ  ASK
00260 0122 09  INX
00270 0123 FF 0128  STX  LAST
00280 0126 20 F1  BRA  READ
00290 0128 0002  LAST  RMR  2
00300          *ENTER COMMAND
00310 012A EF A042 ASK  LDS  #$A042
00320 012D 9D 2D  BSR  CRLF
00330 012F 86 3E  LDA A  #'>

```

Both programs take as their input a cassette tape of the BASIC source program in the format produced by Version 2.0 (and possibly later versions) of the SWTP 8K BASIC. BASEDIT allows editing of the program and then produces a new cassette (or listing) in the same format, so that the program can be fed back into the BASIC interpreter. BASLIST reads a tape and produces a neatly formatted listing separated into pages, each of which is numbered and has an optional program title.

The BASEDIT Program

BASEDIT is a machine-lan-

guage program that lies in locations 0100 through 0497 hex, and uses the rest of memory, from location 0498 hex up through the limit of RAM, to store the BASIC source program to be edited. It has a starting address of 0100 hex, and a 0103 soft-start address that allows BASEDIT to be restarted without erasing the BASIC program from memory.

When BASEDIT is started at the hard-start address of 0100, it immediately goes into its read mode, where all characters entered from the keyboard or tape are stored into memory in the text buffer. Normally, you would start BASEDIT at 0100,

either by loading A048 with 0100 and then typing G with MIKBUG, or typing J 0100 for SWTBUG. BASEDIT then does a carriage return and waits for text. At this point, you should switch to cassette input, and load a standard BASIC program exactly as it was recorded by the BASIC interpreter with its SAVE command.

BASIC programs in this format consist of a chain of 00 (Null) codes, a set of BASIC statement lines, an end code using a 03 code (ETX or Ctrl-C) and another string of 00 (Null) codes. Each line, consisting of 02 (STX or Ctrl-B), a four-digit line number, one space and the BASIC source text, is output by the interpreter in line-number order. This is the format BASEDIT needs.

When the source program is finished reading, turn off the cassette and switch back to keyboard input, and enter a Ctrl-E from the keyboard (hex 05) to signify the end of the program. BASEDIT then goes into the command mode, prints a > and waits for a command.

There are five possible editing commands; each is a single letter (E, P, F, R or S) possibly followed by one or two arguments.

1. E. Typing an E (Exit) returns control to MIKBUG or SWTBUG. BASEDIT leaves an address of 0103 in A048 and A049, so the program can be restarted either by typing G or J 0103. Either way, the program being edited will remain. If a new program is to be entered, then BASEDIT should be restarted at 0100.

2. P. Typing the P command (for Print) outputs the BASIC text as it has been stored or modified. The output begins as soon as the P is entered, without any intervening delays; it is assumed that the null characters read in from the source tape will provide the necessary delays before and after the text.

The P command can be used to simply print out the program, or, if the cassette recorder is on, then the program will be written to cassette in a format suitable for reading into the BASIC system with the LOAD

command. To stop output in the middle of the program, simply press the reset button on the computer's front panel, and then restart from location 0103.

3. F. Typing the letter F (for Find), followed by a string of up to 40 characters, will print out all lines in the text that contain that string. The string must be preceded and followed by a delimiter character, which may be any character on the keyboard except for a carriage return (CR) or any character that appears in the string itself.

For example, the command F.PRINT. prints out all lines that contain the string PRINT; in this case the delimiter is a period. Since any character can be a delimiter, typing FXGOTOX prints out all lines that contain the string GOTO; in this case an X is the delimiter.

The Find command can also be used to print out specific lines or sections of the BASIC program being edited. Since each line of the BASIC program starts with a Ctrl-B character, followed by a four-digit line number, this can be part of the string to be found.

For instance, entering the command F'b0140' (where b stands for Ctrl-B) prints out only line 0140 of the source program. Or, the command F'b05' prints out all lines whose line numbers start with 05, that is, lines 0500 through 0599. This is, of course, most useful if the keyboard can generate the Ctrl-B code; if not, then more lines will be printed. For instance, the command F'0140' prints out not only line number 0140, but also all other lines which contain the string 0140.

4. R. The command R stands for Replace, and it allows any string to be replaced by another string throughout the program. The format for the R command is: R, a delimiter, the first (old) string, the same delimiter, the second (new) string and the same delimiter again. The delimiter follows the same rules as in the F command, and the two strings together may not exceed 40 characters. The two strings need not be the same length; the second string

00340	0131	BD	E1D1	JSR	OUTEEE	
00350	0134	BD	31	BSR	SPACE	
00360	0136	BD	E1AC	JSR	INEEE	
00370	0139	81	50	CMP A	#'P	
00380	0139	26	03	BNE	NOTP	
00390	013D	7E	0201	JMP	PRINT	
00400	0140	81	46	NOTP	CMP A	#'F
00410	0142	26	03	BNE	NOTFIN	
00420	0144	7E	0218	JMP	FIND	
00430	0147	81	52	NOTFIN	CMP A	#'R
00440	0149	26	03	BNE	NOTR	
00450	014B	7E	0246	JMP	REPLAC	
00460	014E	81	53	NOTR	CMP A	#'S
00470	0150	26	03	BNE	NOTS	
00480	0152	7E	02EF	JMP	SEQUEN	
00490	0155	81	45	NOTS	CMP A	#'E
00500	0157	26	D1	BNE	ASK	
00510	0159	7E	E0E3	JMP	CONTRL	
00520				*CR/LF	PRINT	SUBR
00530	015C	86	0D	CRLF	LDA A	#SOD
00540	015E	BD	E1D1	JSR	OUTEEE	
00550	0161	96	0A	LDA A	#S0A	
00560	0163	BD	E1D1	JSR	OUTEEE	
00570	0166	39		RTS		
00580				*PRINT	SPACE	SUBR
00590	0167	86	20	SPACE	LDA A	#S20
00500	0169	BD	E1D1	JSR	OUTEEE	
00610	016C	39		RTS		
00620				*READ	STRING	SUBR
00630	016D	BD	E1AC	READST	JSR	INEEE
00640	0170	11		CBA		
00650	0171	27	09	BEQ	EXITRS	
00660	0173	81	0D	CMP A	#SOD	
00670	0175	27	05	BEQ	EXITRS	
00680	0177	A7	00	STA A	0,X	
00690	0179	08		INX		
00700	017A	20	F1	BRA	READST	
00710	017C	39		EXITRS	RTS	
00720				*FIND	STRING	SUBR
00730	017D	A5	00	FINDST	LDA A	0,X
00740	017F	81	01C4	CMP A	STRING	
00750	0182	27	09	BEQ	FOUND1	
00760	013A	09		FCONT	INX	
00770	0185	BC	0128	CPX	LAST	
00780	0188	26	F3	BNE	FINDST	
00790	018A	C6	00	LDA B	#0	
00800				*B=0	IF NOT FOUND	
00810	018C	39		RTS		
00820				*FOUND	1ST CHAR; CHECK MORE	
00830	018D	FF	01BE	FOUND1	STX	SAVEX
00840	0190	FF	01C0	STX	TEMPSO	
00850	0193	CE	01C4	LDX	#STRING	
00860	0196	FF	01C2	STX	TEMPST	
00870	0199	FE	01C0	FLOOPI	LDX	TEMPSO
00880	019C	A6	00	LDA A	0,X	
00890	019E	03		INX		
00900	019F	FF	01C0	STX	TEMPSO	
00910	01A2	FE	01C2	LDX	TEMPST	
00920	01A5	E6	00	LDA B	0,X	
00930	01A7	08		INX		
00940	01A8	FF	01C2	STX	TEMPST	
00950	01AB	11		CBA		
00960	01AC	26	0B	BNE	NOTF	
00970	01AE	BC	01EC	CPX	SPOINI	
00980	01B1	26	E6	BNE	FLOOPI	
00990	0133	FE	01BE	LDX	SAVEX	
01000	01B6	C6	01	LDA B	#1	
01010				*B=1	IF FOUND	
01020	0138	39		RTS		
01030	01B9	FE	01BE	NOTF	LDX	SAVEX
01040	01BC	20	C6	BRA	FCONT	
01050	01BE	0092		SAVEX	RMB	2
01060	01C0	0002		TEMPSO	RMB	2
01070	01C2	0002		TEMPST	RMB	2
01080	01C4	0028		STRING	RMB	40
01090	01EC	0002		SPOINI	RMB	2
01100	01EE	0002		SPOIN2	RMB	2
01110				*GET	DELIM & STRING	SUBR
01130	01F0	BD	0167	DELIM	JSR	SPACE
01140	01F3	BD	E1AC	JSR	INEEE	
01150	01F6	16		TAB		
01160	01F7	CF	01C4	LDX	#STRING	
01170	01FA	BD	016D	JSR	READST	
01180	01FD	FF	01EC	STX	SPOINI	
01190	0200	39		RTS		
01200				*PRINT	TEXT	
01210	0201	BD	015C	PRINT	JSR	CRLF
01220	0204	CE	0498	LDX	#TEXT	
01230	0207	A6	00	FLOOP	LDA A	0,X
01240	0209	BD	E1D1	JSR	OUTEEE	
01250	020C	08		INX		
01260	020D	BC	0128	CPX	LAST	
01270	0210	26	F5	BNE	FLOOP	
01280	0212	BD	015C	JSR	CRLF	
01290	0215	7E	012A	JMP	ASK	
01300				*FIND	DESIRED	STRING
01310	0218	8D	D6	FIND	BSR	DELIM
01320	021A	9D	0167	JSR	SPACE	
01330	021D	CE	0493	LDX	#TEXT	
01340	0220	BD	017D	FLOOP	JSR	FINDST
01350	0223	5D		TST B		
01360				*QUIT	IF NOT FOUND	
01370	0224	26	03	BNE	LOOKLF	
01380	0226	7E	012A	JMP	ASK	
01390				*BACKSPACE	TO LF	
01400	0229	09		LOOKLF	DEX	
01410	022A	A5	00	LDA A	0,X	
01420	022C	81	0A	CMP A	#S0A	
01430	022E	26	F9	BNE	LOOKLF	

```

01440 0230 9D 015C JSR CRLF
01450 0233 08 INX
01460 *PRINT TO NEXT LF
01470 0234 A6 00 FPRLP LDA A 0,X
01480 0235 91 0A CMP A #504
01490 0238 27 E6 BEQ FLOOP
01500 023A BD E1D1 JSR OUTEEE
01510 023D 08 INX
01520 023E BC 0128 CPX LAST
01530 0241 26 F1 BNE FPRLP
01540 0243 7E 012A JMP ASK
01550 *SUBSTITUTE NEW STRING
01560 0246 9D A3 REPLAC BSR DELIM
01570 0248 BD 016D JSR READST
01580 024B FF 01EE REPL2 STX SPO1V2
01590 024E BD 0167 JSR SPACE
01600 0251 4F CLR A
01610 0252 8D 13 BSR CALCDF
01620 0254 CE 0493 LDX #TEXT
01630 0257 9D 017D RLOOP3 JSR FINJST
01640 025A 5D TST B
01650 025B 27 07 BEQ GOASK
01660 025D 8D 1A BSR SWITCH
01670 025F FE 01BE LDX SAVEX
01680 0262 20 F3 BRA RLOOP3
01690 0264 7E 012A GOASK JMP ASK
01700 *SUB3R TO COMPARE STRING LENGTHS
01720 0267 4A CALCDF DEC A
01730 0269 09 DEX
01740 026B BC 01EC CPX SPO1N1
01750 026C 26 F9 BNE CALCDF
01760 026E 4C RLOOP2 INC A
01770 026F 09 DEX
01780 0270 8C 01C4 CPX #STRING
01790 0273 26 F9 BNE RLOOP2
01800 0275 B7 02E6 STA A COMPAR
01810 0278 39 RTS
01920 *SUBROUTINE TO SWITCH OLD FOR NEW
01930 0279 B6 02E6 SWITCH LDA A COMPAR
01940 027C 27 44 BEQ SAME
01950 027E 2A 21 SPL SHRINK
01960 *NEW STRING LONGER
01970 0280 40 STRETG NEG A
01980 0281 B7 028A STA A MOVETO+1
01990 0284 FE 0128 LDX LAST
01990 0287 E6 00 RLOOP4 LDA B 0,X
01990 0289 E7 00 MOVETO STA B 0,X
01990 028B 09 DEX
01990 028C 8C 01BE CPX SAVEX
01990 028F 26 F6 BNE RLOOP4
01990 0291 BB 0129 ADD A LAST+1
01990 0294 B7 0129 STA A LAST+1
01990 0297 B6 0128 LDA A LAST
01990 029A 99 00 ADC A #0
01990 029C B7 0128 STA A LAST
02000 029F 20 21 BRA SAME
02010 *NEW STRING SHORTER
02020 02A1 B7 02A8 SHRINK STA A MOVEFR+1
02030 02A4 FE 01BE LDX SAVEX
02040 02A7 E5 00 MOVEFR LDA B 0,X
02050 02A9 E7 00 STA B 0,X
02060 02AB 08 INX
02070 02AC BC 0128 CPX LAST
02080 02AF 26 F6 BNE MOVEFR
02090 02B1 B6 0129 LDA A LAST+1
02100 02B4 30 02E6 SUB A COMPAR
02110 02B7 B7 0129 STA A LAST+1
02120 02BA B6 0123 LDA A LAST
02130 02BD 92 00 SBC A #0
02140 02BF 97 0123 STA A LAST
02150 *STRINGS SAME LENGTH
02160 02C2 FE 01EC SAME LDX SPO1N1
02170 02C5 FF 02E7 STX TEMP
02180 02C8 3C 01EE CPX SPO1N2
02190 02CB 26 01 BNE SMLoop
02200 02CD 39 RTS
02210 02CE A6 00 SMLoop LDA A 0,X
02220 02D0 FE 01BE LDX SAVEX
02230 02D3 47 00 STA A 0,X
02240 02D5 03 INX
02250 02D6 FF 01BE STX SAVEX
02260 02D9 FE 02E7 LDX TEMP
02270 02DC 08 INX
02280 02DD FF 02E7 STX TEMP
02290 02E0 9C 01EE CPX SPO1N2
02300 02E3 26 E9 BNE SMLoop
02310 02E5 39 RTS
02320 02E6 0001 COMPAR RMB 1
02330 02E7 0002 TEMP RMB 2
02340 02E9 0004 LINENO RMB 4
02350 02ED 0002 SAVESX RMB 2
02360 *RESEQUENCE LINES STARTING WITH 0010
02370 02EF 9D 015C SEQUEN JSR CRLF
02380 *SET LINE NUMBER TO 0000
02390 02F2 36 30 LDA A #'0
02400 02F4 B7 02E9 STA A LINENO
02410 02F7 B7 02EA STA A LINENO+1
02420 02FA B7 02EB STA A LINENO+2
02430 02FD 97 02EC STA A LINENO+3
02440 0300 CE 0493 LDX #TEXT
02450 *PROCESS EACH LINE OF SOURCE
02460 *LOOK FOR NEXT STX
02470 0303 A6 00 SQLoop LDA A 0,X
02480 0305 81 02 CMP A #502
02490 0307 27 09 BEQ GOTSTX
02500 0309 03 INX
02510 030A BC 0128 CPX LAST
02520 030D 26 F4 BNE SQLoop
02530 030F 7E 047B GODONE JMP DONE

```

may even be what is often called "the null string"—that is, it need not exist. In this case, the first string is simply deleted wherever it occurs.

For example, typing R.THEN .GO TO. simply replaces all occurrences of THEN with the words GO TO. If the old and new strings are the same length, then the replacement, even for long programs, is almost instantaneous; if they are different lengths, then it takes a bit longer. For example, the command R.THEN. removes all THENs and closes up the empty spaces. If there are many occurrences of THEN, then it may take a few seconds for the job to be complete.

The Replace command is very useful for changing messy BASIC programs into neat ones; I have used it often to straighten up programs for publication. For instance, the BASIC statement

```
1000 IFA = 2GOTO30
```

is perfectly acceptable, but difficult to read. It can be changed to

```
1000 IF A = 2 GO TO 30
```

for better appearance.

In a program that has many such statements, it is a simple matter to fix them all with the R command. For example, the following sequence of commands unpacks the GOTO and inserts spaces before and after each.

```
R .GOTO. GO TO .
```

```
R . GO TO. GO TO.
```

```
R .GO TO .GO TO.
```

The first command inserts extra spaces as shown. Just in case some GOTOs already had spaces before or after them and now have two spaces, the last two commands replace any occurrence of GO TO with two spaces either before or after it with just one space. A similar set of commands provides spaces around the IF and equal signs.

Another use for the R command is to change variable names. For example, if the variable N9 is to be replaced by N, use the command F.N9.N.

The F command is extremely powerful, but it does require a bit of care. For example, the total length of the old plus the

new string must be 40 characters or less. If a very long string is to be replaced by another long string, it may sometimes be necessary to first replace the old string with a dummy string such as (), and then replace the string () with the new string. Of course, the F command should first be used to check that the string () did not exist in the original program, or else the result may not be exactly what you wanted!

5. S. The most powerful (and most difficult) command to implement is the S, or Sequence, command. The S command resequences all line numbers of the program to start with the number 0010 and increment in steps of 10.

The Sequence command not only changes the line number of each line, but it also corrects all references to line numbers in the program itself. All IF, GO TO, GOSUB and ON . . . GO TO statements are updated to refer to the new line numbers. Thus, the modified program will run after resequencing. (The S command is pretty smart, but there are two things it will not catch. A reference to a line number in a REMark statement may not be updated correctly, and a line reference inside a PRINT—as in PRINT "NOW EXECUTING LINE 100"—will also not be changed.)

The most common use for the S command is to renumber a program that has been extensively modified to open up the line numbers for further modification. It can also be used by "neatniks" (like me) who want people to think their programs were written right the first time and did not need any additions or deletions! The Sequence command was the main reason for my writing BASEDIT; all other functions are simply by-products. Sequence uses F for finding references in the program to line numbers being changed, and then uses R to change them.

Fig. 1 shows a simple example of using BASEDIT to edit a program. Assuming that a BASIC program has been read from a cassette, we start by printing it with the P command.

```

02540          *GOT A STX
02550 0312 03  GOTSTX INX
02560 0313 BC 0128 CPX LAST
02570 0316 27 F7 BEQ GOJONE
02580 0318 A6 00 LDA A 0,X
02590          *FOR CONTROL CHARACTERS, SKIP
02600 031A 81 30 CMP A #'0
02610 031C 2D F4 BLT GOTSTX
02620          *FOR LETTERS, GO TO NEXT LINE
02630 031E 91 39 CMP A #'9
02640 0320 2E E1 BGT SQLLOOP
02650          *GOT FIRST DIGIT OF LINE NUMBER
02660          *INCREMENT NEW LINE NUMBER
02670 0322 FF 02ED STX SAVESX
02680 0325 CE 02EC LDX #LINENO+3
02690 0328 09 SLOOP2 DEX
02700 0329 A6 00 LDA A 0,X
02710 032B 4C INC A
02720 032C A7 00 STA A 0,X
02730 032E 81 3A CMP A #53A
02740 0330 26 06 BNE LNOK
02750 0332 86 30 LDA A #'0
02760 0334 A7 00 STA A 0,X
02770 0336 2D F0 BRA SLOOP2
02780          *SUBSTITUTE NEW LINE NUMBER
02790 0338 CE 01C4 LNOK LDX #STRING
02800 0339 FF 01EC STX SPOINI
02810 033E CE 02E9 LDX #LINENO
02820 0341 FF 04B9 STX SAVELX
02830 0344 7F 048B CLR ZFLAG
02840 0347 C6 04 LDA B #4
02850 0349 FE 02ED LDX SAVESX
02860 034C A6 00 SLOOP3 LDA A 0,X
02870 034E 7D 048B TST ZFLAG
02880 0351 26 07 BNE SSTORE
02890 0353 81 30 CMP A #'0
02900 0355 27 0C BEQ SSUBST
02910 0357 7C 048B INC ZFLAG
02920 035A FE 01EC SSTORE LDX SPOINI
02930 035D A7 00 STA A 0,X
02940 035F 08 INX
02950 0360 FF 01EC STX SPOINI
02960 0363 FE 0489 SSUBST LDX SAVELX
02970 0366 A6 00 LDA A 0,X
02980 0368 08 INX
02990 0369 FF 0489 STX SAVELX
03000 036C FE 02ED LDX SAVESX
03010 036F A7 00 STA A 0,X
03020 0371 08 INX
03030 0372 FF 02ED STX SAVESX
03040 0375 5A DEC B
03050 0376 26 D4 BNE SLOOP3
03060          *SUBSTITUTED NEW LINE NUMBER FOR OLD
03070          *NOW SEARCH FOR OLD LINE NUMBERS IN TEXT
03080 0378 CE 0498 LDX #TEXT
03090 037B BD 017D SLOOP4 JSR FINDST
03100 037E 5D TST B
03110          *GO ON TO NEXT LINE IF NO REFERENCES
03120 037F 26 06 BNE GOTLN
03130 0381 FE 02ED LDX SAVESX
03140 0384 7E 0303 JMP SQLLOOP
03150          *FOUND REFERENCE TO LINE;
03160          *NEXT MUST NOT BE A DIGIT
03170 0387 FE 01C0 GOTLN LDX TEMPSO
03180 039A A6 00 LDA A 0,X
03190 039C 91 30 CMP A #'0
03200 039E 2D 06 BLT LNOK1
03210 0399 81 39 CMP A #'9
03220 0392 2E 02 BGT LNOK1
03230 0394 20 E5 BRA SLOOP4
03240          *IS IT PRECEDED BY GOTO, GOSUB, OR THEN?
03250 0396 FE 01BE LNOK1 LDX SAVEX
03260 0399 7F 048B CLR ZFLAG
03270 039C 09 LNOK2 DEX
03280 039D A6 00 LDA A 0,X
03290 039F 81 30 CMP A #'0
03300 03A1 27 F9 BEQ LNOK2
03310 03A3 81 20 CMP A #520
03320 03A5 27 F5 BEQ LNOK2
03330          *LOOK FOR COMMA IN ON...GOTO
03340 03A7 81 2C CMP A #','
03350 03A9 26 05 BNE NCOMMA
03360 03AB 7C 048B INC ZFLAG
03370 03AE 20 EC BRA LNOK2
03380          *CHECK IF COMMA EXISTED BEFORE
03390 03B0 7D 048B NCOMMA TST ZFLAG
03400 03B3 27 0E BEQ LASTLT
03410 03B5 81 05 CMP A #505
03420 03B7 27 E3 BEQ LNOK2
03430 03B9 81 30 CMP A #'0
03440 03BB 2D 06 BLT LASTLT
03450 03BD 81 39 CMP A #'9
03460 03BF 2E 02 RGT LASTLT
03470 03C1 20 D9 BRA LNOK2
03480 03C3 FF 048C LASTLT STX SAVERX
03490 03C6 81 4F CMP A #'0
03500 03C8 27 14 BEQ LGOTO
03510 03CA 91 42 CMP A #'B
03520 03CC 27 1E BEQ LGOSUB
03530 03CE 81 4E CMP A #'N
03540 03D0 27 13 BEQ LTHEN
03550          *NO GOTO, THEN, OR GOSUB
03560          *SO KEEP LOOKING
03570 03D2 FE 01BE FAIL LDX SAVEX
03580 03D5 08 INX
03590 03D6 BC 0128 CPX LAST
03600 03D9 26 A0 BNE SLOOP4
03610 03DB 7E 047B JMP DONE
03620          *FOUND AN O, LOOK FOR GOT

```

```

03630 03DE C6 03 LGOTO LDA B #3
03640 03E0 CE 0491 LDX #GOTSTR+3
03650 03E3 20 0C BRA LOOKGO
03660          *FOUND AN N, LOOK FOR THE
03670 03E5 C6 03 LTHEN LDA B #3
03680 03E7 CE 0498 LDX #THENST+3
03690 03EA 20 05 BRA LOOKGO
03700          *FOUND A B, LOOK FOR GOSU
03710 03EC C6 04 LGOSUB LDA B #4
03720 03EE CE 0495 LDX #GOSUBS+4
03730 03F1 FF 0489 LOOKGO STX SAVELX
03740 03F4 FE 048C LDX SAVERX
03750 03F7 09 SLOOP5 DEX
03760 03F8 FF 048C STX SAVERX
03770 03FB A6 00 LDA A 0,X
03780 03FD 81 20 CMP A #520
03790 03FF 27 F6 BEQ SLOOP5
03800 0401 FE 0489 LDX SAVELX
03810 0404 09 DEX
03820 0405 FF 0489 STX SAVELX
03830 0408 A1 00 CMP A 0,X
03840 040A 26 C6 BNE FAIL
03850 040C FE 048C LDX SAVERX
03860 040F 5A DEC B
03870 0410 26 E5 BNE SLOOP5
03880          *DEFINITELY FOUND GOTO, GOSUB, OR THEN
03890          *PUT 05 AND NEW LINE NUMBER IN 'STRING'
03900 0412 FE 01EC LDX SPOINI
03910 0415 86 05 LDA A #505
03920 0417 A7 00 STA A 0,X
03930 0419 08 INX
03940 041A FF 01EE STX SPOIN2
03950          *SKIP ZEROES AT START
03960 041D CE 02E9 LDX #LINENO
03970 0420 A6 00 SLOOP6 LDA A 0,X
03980 0422 81 30 CMP A #'0
03990 0424 26 03 BNE SLOOP7
04000 0426 08 INX
04010 0427 20 F7 BRA SLOOP6
04020          *GOT FIRST NONZERO DIGIT
04030 0429 FF 0489 SLOOP7 STX SAVELX
04040 042C A6 00 LDA A 0,X
04050 042E FE 01EE LDX SPOIN2
04060 0431 A7 00 STA A 0,X
04070 0433 09 INX
04080 0434 FF 01EE STX SPOIN2
04090 0437 FE 0489 LDX SAVELX
04100 043A 08 INX
04110 043B 8C 02ED CPX #LINENO+4
04120 043E 26 E9 BNE SLOOP7
04130          *90TH LINE NUMBERS IN 'STRING'
04140          *FIND DIFFERENCE IN LENGTHS
04150 0440 FE 01EE LDX SPOIN2
04160 0443 4F CLR A
04170 0444 BD 0267 JSR CALCDF
04180          *GO TO DO SWITCH
04190 0447 BD 0279 JSR SWITCH
04200          *AFTER REPLACEMENT OF LINE NUMBERS,
04210          *UPDATE CURRENT POINTER IF STRINGS
04220          *WERE DIFFERENT LENGTHS AND REFERENCE
04230          *WAS BEFORE CURRENT LINE
04240 044A F6 02E6 LDA B COMPAR
04250 044D 26 03 BNE CHECK1
04260 044F 7E 03D2 GOFAIL JMP FAIL
04270          *DIFFERENT LENGTHS; BEFORE?
04280 0452 B6 02ED CHECK1 LDA A SAVESX
04290 0455 B1 01BE CMP A SAVEX
04300 0458 27 04 BEQ CHECK2
04310 045A 2A 0A BPL UPDATE
04320 045C 20 F1 BRA GOFAIL
04330 045E B6 02EE CHECK2 LDA A SAVESX+1
04340 0461 B1 01BF CMP A SAVEX+1
04350 0464 29 E9 BNE GOFAIL
04360          *YES, SO UPDATE
04370 0466 FE 02ED UPDATE LDX SAVESX
04380 0469 5D TST B
04390 046A 2A 09 BPL SUBPTR
04400 046C 03 ADDPTR INX
04410 046D 5C INC B
04420 046E 26 FC BNE ADDPTR
04430 0470 FF 02ED PTROK STX SAVESX
04440 0473 20 DA BRA GOFAIL
04450 0475 09 SUBPTR DEX
04460 0476 5A DEC B
04470 0477 26 FC BNE SUBPTR
04480 0479 20 F5 BRA PTROK
04490          *WHEN DONE REMOVE ALL 05 CODES
04500 047B 86 05 DONE LDA A #505
04510 047D B7 01C4 STA A STRING
04520 0480 CE 01C5 LDX #STRING+1
04530 0483 FF 01EC STX SPOINI
04540 0486 7E 024B JMP REPL2
04550 0489 000? SAVELX RMB 2
04560 048B 0001 ZFLAG RMB 1
04570 048C 0002 SAVERX RMB 2
04580 048E 47 GOTSTR FCC 'GOT'
04590 0491 47 GOSUBS FCC 'GOSU'
04590 0492 4F
04590 0493 53
04590 0494 55
04600 0495 54 THENST FCC 'THE'
04600 0496 48
04600 0497 45
04610 0498 0001 TEXT RMB 1
04620 A043 ORG #A048
04630 A048 0100 FDB #A048
04640          END

```

Then the command R.GOTO.
GO TO. Inserts a space into all

```
> P
0001 INPUT A
0002 ON A GOTO 10,100
0007 GO TO 1
0010 IF A=A*2 THEN 2000
0100 GOSUB 5000
5000 PRINT "SPOT"
5001 END

> R .GOTO.GO TO.
> R .SPOT.STOP.
> R .....
> F .PRINT.
5000 PRINT "STOP"
> S

> P
0010 INPUT A
0020 ON A GO TO 40, 50
0030 GO TO 10
0040 IF A=A*2 THEN 2000
0050 GOSUB 60
0060 PRINT "STOP"
0070 END
```

Fig. 1. Sample printout produced by BASEDIT.

GOTOs. The command R.SPOT .STOP changes the spelling of STOP. The command R.,, . adds a space after every comma. A Find command finds all lines containing the word PRINT; the S command renumbers the lines; and finally, the P command prints out the result.

The BASLIST Program

BASLIST is not nearly as useful as BASEDIT, but it, too, is handy to have. BASLIST accepts as input a BASIC source program cassette, splits it up into 11-inch pages of 66 lines and numbers each page. It is also a machine-language program occupying locations 0100 through 01A1 in memory; the remaining memory from 01A2 and up is used for storing the BASIC program to be printed. Its hard-start address is 0100, and the restart address is 0103.

When BASLIST is started at

0100, it immediately goes into its read mode. A program can then be read in from cassette tape and stored into memory by BASLIST. When the cassette is done, the paper in the printer should be positioned about four lines from the bottom of a page—assuming that perforated paper is used. (BASLIST assumes that the printer is a teleprinter or other serial printer; it is not designed to work with the SWTP printer.)

Once the paper is properly positioned, start the printout by typing a Ctrl-E (hex 05) code on the keyboard. The program then does eight line feeds and starts printing. When printing is over, BASLIST goes into a loop and halts. It can be interrupted by pressing the reset button; the printout can be restarted from the beginning at any time by starting at 0103.

Many BASIC tapes from the

SWTP 8K BASIC system are preceded by the word SAVE and followed by an extra READY. BASLIST removes these from the printout and provides a neat listing of only the program itself. (This is handy for preparing BASIC programs for submission to *Kilobaud*.)

Each page of the printed output is preceded by the words PAGE NO. and a page number starting at 1. The text for this line is stored at locations 0124 through 0136 of the program (see line 00270 of the BASLIST source listing). But the words PAGE NO. are preceded by ten spaces in locations 0124 through 012D. Or, if desired, these ten spaces could be replaced by the ASCII codes for up to nine characters of a program name prior to running (using the M command of the monitor); the program name will then also print on every page. ■

```
00010          NAM      BASLIST
00020          *PROGRAM TO PAGINATE BASIC LISTINGS
00030          OPT      0
00040          *MONITOR ENTRY POINTS
00050          EIAC     INEE     EQU     $EIAC
00060          EID1     OUTEEE    EQU     $EID1
00070          EOTE     PDATA     EQU     $EOTE
00080          E0BF     OUTBYT    EQU     $E0BF
00090 0100        ORG      $0100
00100          *HARD AND SOFT STARTS
00110 0100 7E 0106 HARDST JMP      START
00120 0103 7E 013E SOFTST JMP      LIST
00130 0106 CE 0103 START  LDX     #$0103
00140 0109 FF A048      STX     $A048
00150 010C BE A042      LDS     #$A042
00160 010F CE 01A2      LDX     #TEXT
00170 0112 BD EIAC LOOP  JSR     INEE
00180 0115 A7 00        STA     0,X
00190 0117 81 05        CMP     A,$05
00200          *EXIT ON CONTROL E
00210 0119 27 23        BEQ     LIST
00220 011B 08          INX
00230 011C FF 0121      STX     LAST
00240 011F 20 F1        BRA     LOOP
00250 0121 0002        LAST  RMB     2
00260 0123 0001        PAGENO RMB     1
00270 0124 20          PAGE   FCC
                                PAGE NO.
0125 20
0126 20
0127 20
0128 20
0129 20
012A 20
012B 20
012C 20
012D 20
012E 50
012F 41
0130 47
0131 45
0132 20
0133 4E
0134 4F
0135 2E
0136 20
00280 0137 04          FCB     $04
00290 0138 0D          CRLFLF FCB     $D,$A,$A,$A
0139 0A
013A 0A
013B 04
00300 013C 0002        XTEMP  RMB     2
00310 013E CE 01A2 LIST  LDX     #TEXT
00320 0141 BE A042      LDS     #$A042
```

```
00330          *SKIP TO FIRST STX 02
00340 0144 A6 00        LOOK   LDA     A,0,X
00350 0146 81 02        CMP     A,$02
00360 0148 27 03        BEQ     GO
00370 014A 08          INX
00380 014B 20 F7        BRA     LOOK
00390          *READY TO PRINT
00400 014D 7F 0123     GO      CLR     PAGENO
00410 0150 86 0D        LDA     A,$0D
00420 0152 BD E1D1     JSR     OUTEEE
00430          *SKIP 8 LINES
00440 0155 C6 08        NEWPAG LDA  B,$8
00450 0157 86 0A        SKIP8  LDA  A,$0A
00460 0159 BD E1D1     JSR     OUTEEE
00470 015C 5A          DEC     B
00480 015D 26 F8        BNE     SKIP8
00490          *PRINT NAME AND PAGE
00500 015F FF 013C     STX     XTEMP
00510 0162 CE 0124     LDX     #PAGE
00520 0165 BD E07E     JSR     PDATA
00530          *INCREMENT AND PRINT PAGE NO
00540 0168 B5 0123     LDA     A,PAGENO
00550 016B 8B 01        ADD     A,$01
00560 016D 19          DAA
00570 016E B7 0123     STA     A,PAGENO
00580 0171 CE 0123     LDX     #PAGENO
00590 0174 BD E0BF     JSR     OUTBYT
00600          *PRINT CR AND 2 LF
00610 0177 CE 0138     LDX     #CRLFLF
00620 017A BD E07E     JSR     PDATA
00630 017D FE 013C     LDX     XTEMP
00640 0180 C6 38        LDA     B,$36
00650          *DO NEW LINE
00660 0182 A6 00        NEWLIN LDA  A,0,X
00670 0184 81 0A        CMP     A,$0A
00680 0186 27 0F        BEQ     ENDLIN
00690 0188 81 03        CMP     A,$03
00700 018A 27 14        BEQ     STOP
00710 018C BD E1D1     JSR     OUTEEE
00720 018F 08          INX
00730 0190 BC 0121     CPX     LAST
00740 0193 27 0B        BEQ     STOP
00750 0195 20 EB        BRA     NEWLIN
00760 0197 BD E1D1     ENDLIN JSR     OUTEEE
00770 019A 08          INX
00780 019B 5A          DEC     B
00790 019C 27 B7        BEQ     NEWPAG
00800 019E 20 E2        BRA     NEWLIN
00810 01A0 20 FE        STOP   BRA     STOP
00820 01A2 0001        TEXT  RMB     1
00830 A043             ORG     $A048
00840 A048 0100        PC      FDB     HARDST
00850             END
```

Program 2. BASLIST program.

OUTPUT

04C9	CE	8000	GOTO
CC	FF	A00A	PORT #0
CF		8603	RESET
D1		A700	THE
D3		8611	ACIA
D5		A700	PORT
D7		8D24	BSR NULLS
D9	BD	015C	JSR CR/LF
DC	CE	0507	LDX TEXT
DF		A600	LDA A 'X' 00
E1	BD	E1D1	JSR OUTEE
E4		08	INX
E5	BC	0128	CPX LAST #
E8		26F5	BNE
EA	BD	015C	JSR CR/LF
ED		8DOE	BSR MORE NULLS
EF		8603	LDA A # 05
F1	BD	E1D1	JSR OUTEE
F4	CE	8004	GOTO
F7	FF	A00A	PORT #1
FA	7E	012A	JMP TO BEGIN
FD		C632	LDA B# 50
FF		4F	CLR A
0500	BD	E1D1	OUTPUTEE
0503		5A	DEC B
0504		26F9	BNE
0506		39	RTS

0507

010F 7E 0498

INPUT

0498 CE 8000

FF A00A

8603

A700

8611

04A4 A700

A6 860A

A8 B7 0507

AB CE 0508

AE BD E1AC

B1 4D

2 27FA

B4 A700

6 8103

B8 2706

BA 08

BB FF 0128

BE 20EE

C0 CE 8004

C3 FF A00A

C6 7E 012A

READ (INPUT ASCII)

TST 00?

BEQ (-05)

STA A IX 00

CMP A 03? ETX

BEQ RETURN CRT

INX

STX LAST#

BRA INPUT

RETURN

TO CRT

GO TO BEGIN

△ LOCATION 0206 FF 050)

021F

0256

0302

037A

0-50

~~△ INPUT TO CRT △ 0110 TO CE 8004~~

△ OUTPUT TO PRINTER △ 04C9 TO CE 801C

△ 0112