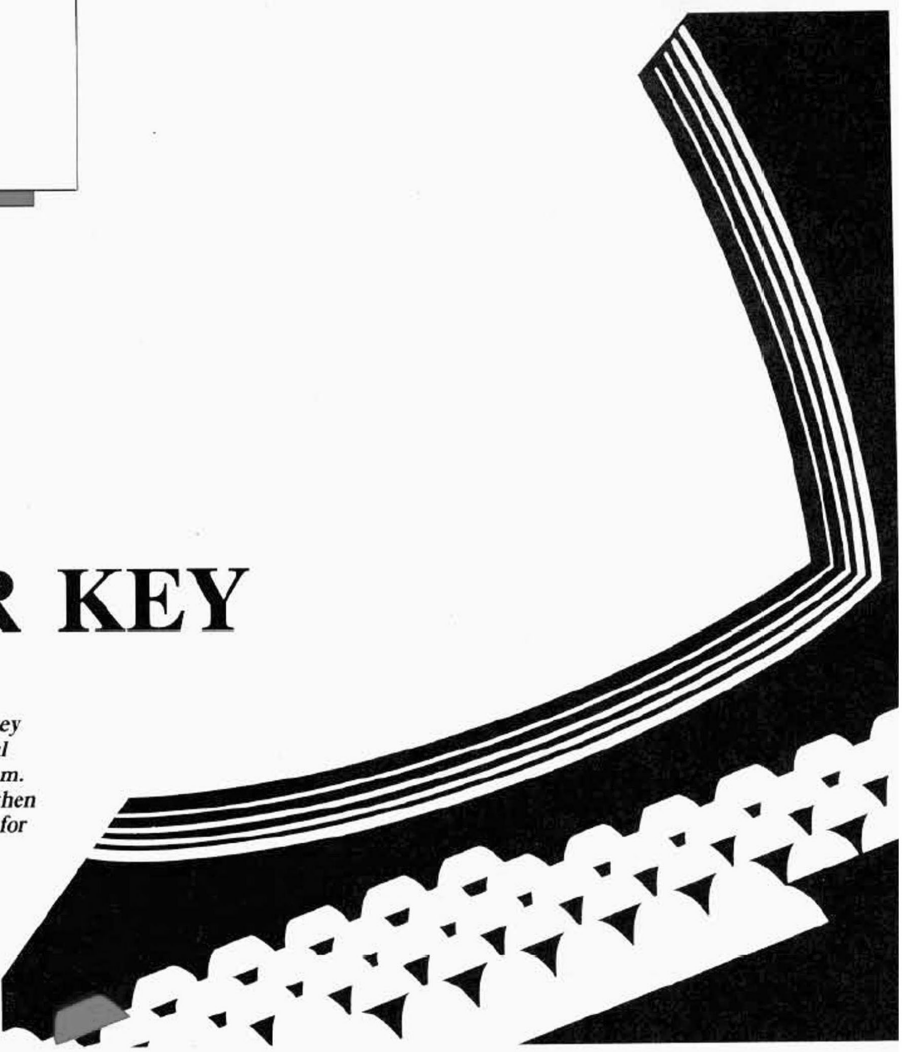
SECOND FEATURE

POWER KEY



DOS 3.3

Define your own single key macros with this powerful machine language program. Your personal macro library can then be stored on disk under DOS 3.3 for easy access.

by Joe Brooks, 521 Moody, Athens, TX 75751

I have often wished that the Apple computer came with user-defined keys. Just think of the time you could save and the keystrokes you could eliminate with such a feature. With just one keystroke, you could type CATALOG, BLOAD, or one of many other frequently used words. It was this wish that led to POWER KEY.

POWER KEY turns the Apple's keys into user-defined keys activated by the <ESC> key. For example, <ESC>C can easily be programmed to type CATALOG (with or without an automatic carriage return), and ESC<N> to type NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY. POWER KEY also adds automatic line numbering to Applesoft BASIC.

USING THE PROGRAM

To use POWER KEY, type **BRUN POWER.KEY**. You may do this either before or after you load an Applesoft program. However, in Integer BASIC, any program previously in memory will be lost. If you should accidentally hit the <RESET> key, type **CALL -29434** to restore POWER KEY. Also, if you switch from one BASIC to the other, type **CALL -29434** or BRUN POWER.KEY to reset HIMEM. POWER KEY automatically loads the data file PGM.DATA, which contains the key assignment information. This file can be automatically updated by POWER KEY at your request.

> "POWER KEY turns the Apple's keys into user-defined keys activated by the <ESC> key."

When you first use POWER KEY, the program buffer contains the entries shown in **Table 1** (the <R> represents an automatic carriage return).

User-defined words are activated by pressing the <ESC> key and typing the desired key. For example, <ESC>C will generate CATOLOG,D1.

POWER KEY also has the following functions:

AUTO<START>,

<INCR>	Auto line numbering
<CTRL>X	Cancel automatic line numbering
<CTRL>P	Program a key
<CTRL>D	Delete a programmed key
<CTRL>L	List programmed keys
<CTRL>Z	Save PGM.DATA file to disk

AUTOMATIC LINE NUMBERING

POWER KEY adds an automatic line numbering feature to Applesoft (Integer BASIC already has this feature). To activate it, type **AUTO (START),(INCR)** where (START) is the starting line number and (INCR) is the line number increment. (This is the same syntax as Integer BASIC's Auto Line Number.) To cancel Auto Line Numbering, type <CTRL>X.

PROGRAMMING A KEY

To program a key, press the <ESC> key, then <CTRL>P. The program will respond with the prompt "PRESS KEY TO BE PROGRAMMED:". If, for some reason, you do not wish to program a key, press the <ESC> key to exit.

Any key, except for ten reserved keys, may be programmed. The reserved keys are:

<CTRL>X
<CTRL>P
<CTRL>D
<CTRL>L
<CTRL>Z
I, J, K, M
<CTRL><SHIFT>P

If you attempt to program one of these keys, the message RESERVED KEY will be printed on the screen, and the program will exit to BASIC. Also, if you press a key that is already programmed, the message KEY IS PROGRAMMED is displayed and the program exits. Note that although (<CTRL> <SHIFT>P) is not a reserved key, it is not allowed because on the II Plus it would be stored as 00, which the program considers empty buffer space.

After you press a valid key (including the control keys and the shift keys), POWER KEY prompts "INPUT PROGRAM DATA:". You may now enter any data of any length you desire. (If you fill the 1K buffer, the message BUFFER FULL will be displayed. Any control character that you type will be printed in inverse text. When you have finished programming the key, you may press either <RETURN> or <CTRL>R. Typing <CTRL>R places an encoded carriage return after the data, activating the Automatic Carriage Return feature.

Next, the program prompts SAVE TO DISK? (Y/N). If you wish to save the updated buffer, type Y for yes. If you press any other key, the program will exit to BASIC. Normally, the buffer only needs to be saved during programming or after deleting a key, but, if you should decide that you want to save the buffer at any time, type <ESC><CTRL>Z.

DELETING A PROGRAMMED KEY

To delete a key that has been programmed, press the <ESC> key, then <CTRL>D. You will be prompted "PRESS KEY TO CLEARED:". When you press the programmed key, it clears instantly. If the key you press is not programmed, the message KEY NOT PROGRAMMED is displayed and POWER KEY exits to BASIC. If you should change your mind about deleting a key, press the <ESC> key to exit.

After deleting a key, the program prompts SAVE TO DISK? (Y/N), allowing you to save the updated buffer if you wish.

Table 1:
Key Assignments

	B	BSAVE
<CTRL>	B	BLOAD
	C	CATALOG,D1<R>
<CTRL>	C	CATALOG,D2<R>
	D	DELETE
	F	FOR I=
<CTRL>	F	FLASH
	G	GOTO
<CTRL>	G	GOSUB
	H	HOME
<CTRL>	H	HTAB
<CTRL>	I	INPUT
<CTRL>	K	CHR$(
	L	LIST<R>
<CTRL>	M	MID$(
	N	NEXT
<CTRL>	N	NORMAL
	P	PRINT
	R	RETURN
<CTRL>	R	RIGHT$(
	S	SAVE
<CTRL>	S	STR$(
	T	TEXT
<CTRL>	T	THEN
	V	VTAB
	<	LEFT$(
	^	INVERSE

LISTING PROGRAMMED KEYS

To list the programmed keys, press <ESC> and then <CTRL>L. Note that control characters are printed in inverse text, and that <CTRL>R appears after the keys with Auto Carriage Return. If the screen is filled, a prompt tells you to "<PRESS ANY KEY TO CONTINUE>". When the listing is complete, the number of free bytes remaining in the buffer is displayed.

ENTERING THE PROGRAM

To key in POWER KEY, enter the machine language program shown in **Listing 1**. If you have an assembler, use it to enter and assemble the source code shown. For direct memory entry, type **CALL-151** to enter the Monitor, then begin entering the machine language code. (For help in entering machine language code, see "A Welcome to New *Nibble* Readers" at the beginning of this issue.) After the program is entered, save it on disk by typing:

BSAVE POWER.KEY,A$8CF0,L$50E

Next, enter the key assignment data shown in **Listing 2.** First we must clear the buffer area in memory to all zeros. To zero the buffer, first, enter the Monitor by typing CALL-151, and type **9200:00**. Now type

9201<9200.95FEM

The entire memory area from $9200 to $95FE (hex) is now all zeros.

Enter the PGM.DATA memory listing, beginning at $9200. After it is entered, set a pointer at the end of the buffer by typing **95FE:AF 92**. To save the listing to disk, type:

BSAVE PGM.DATA,A$9200,L$400

Now type <CTRL> C to re-enter BASIC.

Note: If you prefer not to use the pre-programmed keys, omit typing in and saving the PGM.DATA buffer file. This will cause a FILE NOT FOUND error message when POWER.KEY is BRUN the first time, but will not affect operation in any other way. The data buffer will contain all zeros, and you can program your own keys from scratch.

Because of the changes Apple has made in the way the //c interprets the <ESC> key, //c owners should perform the following steps after entering and saving the listing:

1. **BLOAD POWER.KEY**
2. **POKE 36381,154:POKE 36421,96**
3. **BSAVE POWER.KEY.//C, A$8CF0,L$50E**

To activate programmed keys, //c owners will then press <CTRL>Z instead of <ESC>. For example, rather than pressing <ESC>C for a catalog, //c owners would press <CTRL>ZC

HOW THE PROGRAM WORKS

POWER KEY is 1294 bytes in length and is located directly below DOS. HIMEM is set below POWER KEY at $8CF0 to protect it from BASIC. The program uses a 1K buffer from $9200 to $95FC. The last two bytes are reserved for saving the storage pointer to disk, and one byte is always 00 to mark the end of the buffer.

POWER KEY consists of eight sections plus the tables at the end of the program. These sections are described in detail below.

Initialize

The Initialize section **(line 1540)** is only used when POWER KEY is first loaded into memory using the BRUN command. First, the data buffer is zeroed. HIMEM is set below the program to protect it from BASIC. A check is made to see which BASIC has been selected, and the appropriate HIMEM locations are set. Next, the keyboard input switch (KSW) is set to point to START. A jump to DOS lets it know about the changes.

Since the KSW points to START, program control returns there. The first time through,

a jump is made to BLDATA (**line 1860**), where the PGM.DATA file is loaded from disk.

The data is BLOADED by transferring the string BLOAD PGM.DATA into the keyboard buffer as though it were typed in from the keyboard. The X-Register must be set to the length of the string. Now, an RTS and DOS do the work! This is an easy way to both save and load disk files within a machine language program.

Before the program exits, the jump to BLDATA at START is replaced with PHA, LDA INTCK. After the PGM.DATA file is loaded, POWER KEY is initialized and ready to go.

Auto Line Number

Auto Line Number (**line 2020**)is the actual start of the program, since the KSW now points to START. First, a check (LDA INTCK) is made to see if we are in Integer BASIC. If so, program flow jumps to the Analyze Input section (**line 3040**). This section is also skipped if the X-Register is not zero, which means that something has been typed into the keyboard buffer. If the keyboard buffer is empty, the program checks to see if the Auto Line Number flag (AUTFLG) is set to -1.

If the flag is set, then the current line number (LINUM) is stored at DSCTMP. LINUM is then incremented by the value stored at LINCR. The incremented LINUM will not be used at this time, but it is now ready for the next pass through.

The Applesoft ROM routines CHNG, FOUT, STRLIT, and FREFAC convert the line number at DSCTMP (now in hex) to a decimal string. This string (pointed to by INDEX) is moved into the keyboard buffer and printed, followed by a space. The line number has now been "typed" into the keyboard buffer. The X-Register is set to the length of the line number string plus one space. The program is now ready for input.

If the AUTFLG is not set, the program branches to SETAUT (**line 2530**) to wait for input (JSR KEYIN). KEYIN is the Monitor ROM keyboard entry routine, which returns with the ASCII value of the key pressed in the Accumulator. The input is compared to the string AUTO. If AUTO has not been typed, the program branches to the next section. If it has been typed, then the line number and increment are input and stored in the keyboard buffer beginning at STEPS.

Next, the input is evaluated at EVAL (**line 2790**). The Applesoft routines CHRGOT and LINGET convert the line number to hex, which is then stored at LINUM. A check is made for a comma. If the line number is not followed by a comma, the program exits.

Next, the Applesoft ROM routines CHRGET and LINGET convert the increment to hex, and it is stored at LINCR. Finally, the Auto Line Number flag is set, and the program exits.

Analyze Input

Analyze Input (**line 3040**) is the main keyboard input routine. To get the keyboard input, the program does a JSR KEYIN. After the KEYIN routine returns, POWER KEY checks for the <ESC> key. If it has not been entered, POWER KEY checks for <CTRL>X, which clears AUTFLG, and then exits with an RTS. If it finds <ESC>, then the program gets another keypress. This time JSR KEY is used. KEYIN is not used here because it does not leave a flashing cursor; the RDKEY routine in the Monitor ROM is not used because it now points to the START of this program (using the KSW switch). Instead, the subroutine KEY places the flashing cursor and then uses KEYIN. After the keypress is made, the program checks to see if it is one of the four reserved keys, I, J, K, or M. If so, the appropriate cursor move is made, and the program loops to get another keypress at KEY.

If the keypress is not I, J, K or M, then POWER KEY checks to see if it is <CTRL>P, D, L or Z. If it is one of these, a branch is made to the appropriate section of the program. If it is not, the input is checked to see if it is a programmed key. If so, then it is printed in the next section of the program. If not, POWER KEY exits with an RTS.

Print Programmed Data

The Print Programmed Data subroutine (**line 3510**) prints the stored data to the screen. It also transfers the data into the keyboard buffer, to simulate typing it in. The X-Register is set to the length of the data. The subroutine PRTCHR is used throughout POWER KEY. This subroutine converts control characters so that they will be printed in inverse text.

When the program exits with an RTS, the character in the Accumulator is printed. To get the proper data printed to the screen and the proper data in the keyboard buffer, the program prints the character under the cursor before exiting, and then the Accumulator is loaded with a back arrow. Now when the program exits, the printed character is backed over, and everything functions normally.

Program Keys

In this section (**line 3830**), the key to be programmed and the program data are input. We are first prompted with the message INPUT KEY TO BE PROGRAMMED. If the input key is a reserved key or if it has already been programmed, the program exits with the appropriate message, RESERVED KEY or KEY IS PROGRAMMED. The programmed key is stored in sequential order, as is the data, but first it is "marked" by clearing bit seven at LEGAL.

We are now prompted with the message "INPUT PROGRAM DATA:". As the data is typed in, several checks are made before POWER KEY stores it in the program buffer.

To allow the use of the back arrow without storing its ASCII code in the data buffer, a check is made for the back arrow at NXCHR. The back arrow routine begins at BCKARR. Here, the data in the buffer is deleted as the back arrow is used by decrementing the storage pointer (STOR), which points to the memory location at which the data will be stored, and then zeroing the data at that location. The appropriate character is also deleted on the screen in the BCKARR routine. A check is made to see if we have backed up too far. If so, the storage pointer and horizontal position are restored, as though no key had been pressed,at the routine TOOFAR.

Checks are also made for the forward arrow, which is not used in this routine. Also, the <RETURN> key is checked, which will cause a branch to the Save to Disk subroutine.

Save to Disk

The Save to Disk subroutine (**line 4800**) saves key assignment data on disk in the file PGM.DATA. The contents of the storage pointer (TEMSTO at $95FE) are also saved so that future programming will be stored at the correct address. The storage pointer is reset to this value at the beginning of the Analyze Input section.

The same method is used to save the data to disk as was used to BLOAD (described in the Initialize section), except that the string BSAVE PGM.DATA,A$9200,L$400 is transferred into the keyboard buffer.

Delete a Programmed Key

This portion of POWER KEY (**line 5060**) first prompts PRESS KEY TO BE CLEARED, and then awaits a keypress (JSR KEY). A check for the <ESC> key is made (pressing <ESC> exits the program).

Next, the subroutine FIND checks to see if the key is programmed. If the key is not programmed, a branch is made to NOTPGM, which prints the message KEY NOT PROGRAMMED and then exits.

The FIND subroutine will return with the read pointer set at the address of the key to delete. The storage pointer is set to this address. Next, the read pointer is incremented until it finds (points to) the next programmed key (bit seven cleared) or a 00, which is the end of the data.

With the pointers set, we are now ready to compress the data by filling in the deleted area with the data that was above it. This is done beginning at STEP19. Finally, the area from which the data was moved must be zeroed. This is done beginning at LAST.

List Programmed Keys

This section (**line 5580**) prints the programmed keys and the corresponding user-defined words, as well as the number of free bytes remaining in the buffer.

After the read pointer is set to the beginning of the buffer, and the screen is cleared, a print-out loop begins at STEP21. In each pass through the loop, the character pointed to by READ is found and checked. If it is a 00, then there is no more data and the number of free bytes is printed. If it is data, it is printed beginning at CHR, the read pointer is incremented, and the program branches to STEP21 for the next pass through the loop.

If it is a programmed key, a check is made to see if the screen is full. If so, the program prints PRESS ANY KEY TO CONTINUE, waits for a keypress, and then clears the screen. The next key assignment is then printed, followed by a dash. The program then continues looping starting at CHR. The number of free bytes in the buffer is printed before POWER KEY exits the routine. The free space which is the difference between the storage pointer and the top of the buffer, minus the three reserved bytes, is calculated. A check is made to see if Integer BASIC is selected. The decimal printout for Integer is INPRT ($E51B) and for Applesoft is LINPRT ($ED24).

LISTING 1: POWER.KEY

```
                1000 ************************
                1010 *                      *
                1020 *     POWER.KEY        *
                1030 *                      *
                1040 *     BY  JOE BROOKS   *
                1050 *     COPYRIGHT 1985   *
                1060 *     BY MICROSPARC,INC.*
                1070 *     CONCORD, MA 01742*
                1080 *                      *
                1090 *     S-C ASSEMBLER    *
                1100 ************************
                1110          OR $8CF0
                1120          TA $800
0006-           1130 STOR     EQ $06        DATA STORAGE POINTER
0008-           1140 READ     EQ $08        DATA READ POINTER
0018-           1150 TEMP     EQ $18        KBD INPUT STORAGE
0019-           1160 FLG      EQ $19        RESTORE POINTER FLAG
001A-           1170 LINCR    EQ $1A        LINE# INCREMENT
00EE-           1180 LINUM    EQ $EE        LINE# STORAGE
0024-           1190 HPOS     EQ $24        CURSOR HOR. POSITION
0025-           1200 VPOS     EQ $25        CURSOR VERT. POSITION
0028-           1210 BASL     EQ $28        CURSOR BASE ADDRESS
0038-           1220 KSWL     EQ $38        KBD INPUT SWITCH (LOW)
0039-           1230 KSWH     EQ $39        KBD INPUT SWITCH (HIGH)
004C-           1240 INTMEM   EQ $4C        HIMEM (INTEGER)
0050-           1250 LINNUM   EQ $50        GEN. PURP. 16 BIT #
005E-           1260 INDEX    EQ $5E        STRING MOVE POINTER
0073-           1270 HIMEM    EQ $73        HIMEM (APPLESOFT)
009E-           1280 DSCTMP   EQ $9E        STRING DESCRIPTOR
00B1-           1290 CHRGET   EQ $B1        GET TEXT FROM TXTPTR
00B7-           1300 CHRGOT   EQ $B7        GET TEXT, NO INC.
00B8-           1310 TXTPTR   EQ $B8        TEXT POINTER
00CA-           1320 PPTR     EQ $CA        INTEGER PROGRAM PNTR.
0200-           1330 KEYBRD   EQ $200       KEYBOARD BUFFER
9200-           1340 BUFFER   EQ $9200      BUFFER START
95FE-           1350 TEMSTO   EQ $95FE      SAVE STOR PNTR HERE
9600-           1360 BUFEND   EQ $9600      BUFFER END
A851-           1370 DOS      EQ $A851      DOS HOOKS
AA5B-           1380 YSTOR    EQ $AA5B      DOS Y REG. STORAGE
DA0C-           1390 LINGET   EQ $DA0C      LINE# FROM TXTPTR
E001-           1400 INTCK    EQ $E001      INTEGER CHECK
E3E7-           1410 STRLIT   EQ $E3E7      STORE QUOTATION
E51B-           1420 INTPRT   EQ $E51B      DEC.# IN INTEGER
E600-           1430 FREFAC   EQ $E600      FREE TEMP. DESCRIPTOR
EBA0-           1440 CHNG     EQ $EBA0      CONVERT NUMBER
ED24-           1450 LINPRT   EQ $ED24      DEC.# IN APPLESOFT
ED34-           1460 FOUT     EQ $ED34      CREATE STRING
FC2C-           1470 ESC1     EQ $FC2C      ESC A,B,C & D
FC58-           1480 HOME     EQ $FC58      CLEAR SCREEN
FD1B-           1490 KEYIN    EQ $FD1B      READ APPLE'S KEYBOARD
FD8B-           1500 CROUT1   EQ $FD8B      <CR> WITH CLEAR
FDED-           1510 COUT     EQ $FDED      CHARACTER OUTPUT
FF3A-           1520 BELL     EQ $FF3A      SOUND BELL
                1530 *
                1540 * INITIALIZE
                1550 *
8CF0- 20 06 91  1560 INIT     JSR SET       SET STORAGE POINTER
8CF3- A8        1570          TAY
8CF4- 91 06     1580 STEP1    STA (STOR),Y  ZERO THE BUFFER
8CF6- C8        1590          INY
8CF7- D0 FB     1600          BNE STEP1
8CF9- E6 07     1610          INC STOR+1
8CFB- A6 07     1620          LDX STOR+1
8CFD- E0 96     1630          CPX /BUFEND   END OF BUFFER?
8CFF- D0 F3     1640          BNE STEP1     NO->CONTINUE
8D01- 20 06 91  1650          JSR SET       SET STORAGE POINTER
8D04- 85 19     1660          STA FLG       CLEAR THE RESTORE FLAG
8D06- AD 01 E0  1670 HMSET    LDA INTCK     INTEGER?
8D09- F0 0A     1680          BEQ INTGER    ->YES
8D0B- A9 F0     1690          LDA #INIT     SET APPLESOFT HIMEM
8D0D- 85 73     1700          STA HIMEM
8D0F- A9 8C     1710          LDA /INIT
8D11- 85 74     1720          STA HIMEM+1
8D13- D0 0C     1730          BNE SETKSW
8D15- A9 F0     1740 INTGER   LDA #INIT     SET INTEGER HIMEM
8D17- 85 4C     1750          STA INTMEM
8D19- 85 CA     1760          STA PPTR
8D1B- A9 8C     1770          LDA /INIT
8D1D- 85 4D     1780          STA INTMEM+1
8D1F- 85 CB     1790          STA PPTR+1
8D21- A9 4E     1800 SETKSW   LDA #START    SET KEYBOARD INPUT
8D23- 85 38     1810          STA KSWL      SWITCH TO
8D25- A9 8D     1820          LDA /START    PROGRAM START
8D27- 85 39     1830          STA KSWH
8D29- 4C 51 A8  1840          JMP DOS       TELL DOS ABOUT IT
                1850 *
8D2C- A2 00     1860 BLDATA   LDX #00       BLOAD DATA
8D2E- BD EC 91  1870 TRSFR    LDA BPD,X
8D31- 9D 00 02  1880          STA KEYBRD,X
8D34- 20 ED FD  1890          JSR COUT
8D37- E8        1900          INX
8D38- E0 0F     1910          CPX #14
8D3A- D0 F2     1920          BNE TRSFR
8D3C- A9 48     1930          LDA #$48      PLACE "PHA
8D3E- 8D 4E 8D  1940          STA START     LDA INTCK"
8D41- A9 AD     1950          LDA #$AD      AT START
8D43- 8D 4F 8D  1960          STA START+1
8D46- A9 01     1970          LDA #01
8D48- 8D 50 8D  1980          STA START+2
8D4B- A9 8D     1990          LDA #$8D      <CR>
8D4D- 60        2000          RTS           ->EXIT
                2010 *
                2020 * AUTO LINE NUMBER
                2030 *
8D4E- 4C 2C 8D  2040 START    JMP BLDATA    ->LOAD DATA FIRST TIME
8D51- E0        2050          .HS E0
                2060 *       -MODIFIES TO:-
                2070 *START   -PHA-         SAVE CHARACTER
                2080 *        -LDA INTCK-   INTEGER?
8D52- F0 04     2090          BEQ JANLYZ    ->YES, SKIP AUTO LN
8D54- E0 00     2100          CPX #00       KBD BUFFER EMPTY?
8D56- F0 04     2110          BEQ STEP2     YES, CHECK AUTFLG
8D58- 68        2120 JANLYZ   PLA
8D59- 4C 19 8E  2130          JMP ANLYZE    ->SKIP AUTO LN
8D5C- AD 0F 91  2140 STEP2    LDA AUTFLG    SHOULD WE AUTO LN?
8D5F- C9 01     2150          CMP #$01
8D61- D0 4A     2160          BNE SETAUT    ->NO
8D63- 68        2170          PLA           RESTORE STACK
8D64- A5 EE     2180          LDA LINUM     INCREMENT THE
8D66- 85 9F     2190          STA DSCTMP+1  LINE NUNBER
8D68- 18        2200          CLC
8D69- 65 1A     2210          ADC LINCR
8D6B- 85 EE     2220          STA LINUM
8D6D- A5 EF     2230          LDA LINUM+1
8D6F- 85 9E     2240          STA DSCTMP
8D71- 65 1B     2250          ADC LINCR+1
8D73- 85 EF     2260          STA LINUM+1
8D75- C9 FA     2270          CMP #$FA      HIGHER THAN 63999?
8D77- 90 03     2280          BCC STEP3     ->NO
8D79- CE 0F 91  2290          DEC AUTFLG    YES, CLEAR AUTO LN FLAG
8D7C- A2 90     2300 STEP3    LDX #$90
8D7E- 38        2310          SEC
8D7F- 20 A0 EB  2320          JSR CHNG      CONVERT THE
8D82- 20 34 ED  2330          JSR FOUT      LINE NUMBER
8D85- 20 E7 E3  2340          JSR STRLIT    TO DIGITS
8D88- 20 00 E6  2350          JSR FREFAC
8D8B- A0 00     2360          LDY #00
8D8D- B1 5E     2370 STEP4    LDA (INDEX),Y MOVE IT TO
8D8F- F0 0B     2380          BEQ STEP5     THE KBD BUFFER
8D91- 99 00 02  2390          STA KEYBRD,Y
8D94- 49 80     2400          EOR #$80
8D96- 20 ED FD  2410          JSR COUT      PRINT IT
8D99- C8        2420          INY
8D9A- D0 F1     2430          BNE STEP4
8D9C- A9 A0     2440 STEP5    LDA #$A0      PRINT AND
8D9E- 99 00 02  2450          STA KEYBRD,Y  STORE A SPACE
8DA1- 20 ED FD  2460          JSR COUT
8DA4- C8        2470          INY
8DA5- 98        2480          TYA
8DA6- AA        2490          TAX
8DA7- 20 C4 90  2500          JSR KEY       NOW GET INPUT
8DAA- 4C 1C 8E  2510          JMP STEP9     ->CONTINUE
                2520 *
8DAD- 68        2530 SETAUT   PLA           RESTORE INPUT
8DAE- 20 1B FD  2540          JSR KEYIN     SHOULD WE
8DB1- 4C B7 8D  2550          JMP STEP7     ACTIVATE AUTO LN?
8DB4- 20 C4 90  2560 STEP6    JSR KEY
8DB7- DD 10 91  2570 STEP7    CMP AUTO,X
8DBA- D0 60     2580          BNE STEP9     ->NO
8DBC- 9D 00 02  2590          STA KEYBRD,X
8DBF- 20 ED FD  2600          JSR COUT
8DC2- E8        2610          INX
8DC3- E0 04     2620          CPX #04
8DC5- D0 ED     2630          BNE STEP6
8DC7- 20 C4 90  2640 STEP8    JSR KEY       YES, INPUT STARTING
8DCA- 20 ED FD  2650          JSR COUT      LINE#, INCREMENT
8DCD- 49 80     2660          EOR #$80
8DCF- 9D 00 02  2670          STA KEYBRD,X
8DD2- C9 0D     2680          CMP #$0D      <CR>?
8DD4- F0 0E     2690          BEQ EVAL      ->YES, NOW EVALUATE
8DD6- C9 08     2700          CMP #$08      BACK ARROW?
8DD8- F0 07     2710          BEQ BACK      ->YES
8DDA- C9 15     2720          CMP #$15      FORWARD ARROW?
8DDC- F0 E9     2730          BEQ STEP8     ->YES
8DDE- E8        2740          INX
8DDF- D0 E6     2750          BNE STEP8
8DE1- CA        2760 BACK     DEX
8DE2- D0 E3     2770          BNE STEP8
                2780 *
8DE4- A9 04     2790 EVAL     LDA #04       GET STARTING LINE#
8DE6- 85 B8     2800          STA TXTPTR
8DE8- A9 02     2810          LDA #02
8DEA- 85 B9     2820          STA TXTPTR+1
8DEC- 20 B7 00  2830          JSR CHRGOT    CONVERT IT
8DEF- 20 0C DA  2840          JSR LINGET    TO HEX
8DF2  A5 50     2850          LDA LINNUM    NOW STORE IT
8DF4- 85 EE     2860          STA LINUM
8DF6- A5 51     2870          LDA LINNUM+1
8DF8- 85 EF     2880          STA LINUM+1
8DFA- 20 B7 00  2890          JSR CHRGOT
8DFD- C9 2C     2900          CMP #$2C      COMMA?
8DFF- D0 11     2910          BNE END       ->NO
8E01- 20 B1 00  2920          JSR CHRGET    GET LINE# INCR.
8E04- 20 0C DA  2930          JSR LINGET    CONVERT IT
8E07- A5 50     2940          LDA LINNUM    STORE IT
8E09- 85 1A     2950          STA LINCR
8E0B- A5 51     2960          LDA LINNUM+1
8E0D- 85 1B     2970          STA LINCR+1
8E0F- EE 0F 91  2980          INC AUTFLG    SET AUTO LN FLAG
8E12- A9 8D     2990 END      LDA #$8D      <CR>
8E14- 60        3000          RTS           ->EXIT
8E15- A2 00     3010 EXIT     LDX #00
8E17- F0 F9     3020          BEQ END
```

```
                3030 *
                3040 * ANALYZE INPUT
                3050 *
8E19- 20 1B FD  3060 ANLYZE   JSR KEYIN       GET INPUT
8E1C- C9 9B     3070 STEP9    CMP #$9B        ESC?
8E1E- F0 08     3080          BEQ INPUT       ->YES
8E20- C9 98     3090          CMP #$98        CTRL-X?
8E22- D0 03     3100          BNE RTS         ->NO
8E24- CE 0F 91  3110          DEC AUTFLG      CLEAR AUTO LN FLAG
8E27- 60        3120 RTS      RTS             EXIT THIS PROGRAM
                3130 *
8E28- A5 19     3140 INPUT    LDA FLG         RESET STORAGE POINTER?
8E2A- D0 11     3150          BNE INPUT1      ->NO
8E2C- AD FF 95  3160          LDA TEMSTO+1    PGM DATA LOADED?
8E2F- F0 0C     3170          BEQ INPUT1      ->NO
8E31- E6 19     3180          INC FLG         YES!
8E33- AD FE 95  3190          LDA TEMSTO      RESET THE POINTER
8E36- 85 06     3200          STA STOR
8E38- AD FF 95  3210          LDA TEMSTO+1
8E3B- 85 07     3220          STA STOR+1
8E3D- 20 C4 90  3230 INPUT1   JSR KEY         GET NEXT INPUT
8E40- 20 8B 90  3240          JSR IJKMCK      I,J,K, OR M?
8E43- 90 0B     3250          BCC STEP10      ->NO
                3260 *
8E45- A8        3270          TAY             ESC IJKM FUNCTIONS
8E46- B9 31 91  3280          LDA TBL-$C9,Y   TRANSLATE IJKM
8E49- 38        3290          SEC             TO CBAD
8E4A- 20 2C FC  3300          JSR ESC1        MOVE CURSOR
8E4D- 4C 3D 8E  3310          JMP INPUT1      ->NEXT INPUT
                3320 *
8E50- C9 90     3330 STEP10   CMP #$90        CTRL-P?
8E52- F0 5A     3340          BEQ PGM         ->YES
8E54- C9 84     3350          CMP #$84        CTRL-D?
8E56- F0 0E     3360          BEQ JDEL        ->YES
8E58- C9 8C     3370          CMP #$8C        CTRL-L?
8E5A- F0 07     3380          BEQ JLIST       ->YES
8E5C- C9 9A     3390          CMP #$9A        CTRL-Z?
8E5E- D0 09     3400          BNE ISKEY       ->NO
8E60- 4C 72 8F  3410          JMP SAVE        ->YES
8E63- 4C 0B 90  3420 JLIST    JMP LIST
8E66- 4C A3 8F  3430 JDEL     JMP DELETE
8E69- 20 9B 90  3440 ISKEY    JSR FIND        IS KEY PROGRAMMED?
8E6C- B0 08     3450          BCS PRTDAT      ->YES
8E6E- A4 24     3460          LDY HPOS        NO, EXIT
8E70- 8C 5B AA  3470          STY YSTOR
8E73- A5 18     3480          LDA TEMP
8E75- 60        3490          RTS
                3500 *
                3510 * PRINT PROGRAMMED DATA
                3520 *
8E76- 20 FF 90  3530 PRTDAT   JSR INCR        INCREMENT READ POINTER
8E79- B1 08     3540          LDA (READ),Y    GET CHARACTER
8E7B- 08        3550          PHP
8E7C- C9 92     3560          CMP #$92        CTRL-R?
8E7E- F0 1C     3570          BEQ AUTOCR      ->YES, DO AUTO <CR>
8E80- 28        3580          PLP
8E81- 10 20     3590          BPL END1        ->LAST ONE
8E83- 9D 00 02  3600          STA KEYBRD,X    PUT IN KEYBOARD BUFFER
8E86- 48        3610          PHA             CHECK FOR FULL KBD BUF
8E87- AD 01 E0  3620          LDA INTCK       INTEGER?
8E8A- D0 04     3630          BNE FP          ->NO
8E8C- E0 7E     3640          CPX #$7E
8E8E- B0 0F     3650          BCS FULL        ->TOO FULL FOR INTEGER
8E90- E0 EF     3660 FP       CPX #$EF
8E92- B0 0B     3670          BCS FULL        ->TOO FULL FOR APPLESOFT
8E94- 68        3680          PLA             KBD BUF'S OK, CONTINUE
8E95- E8        3690          INX
8E96- 20 ED FD  3700          JSR COUT        PRINT IT
8E99- 4C 76 8E  3710          JMP PRTDAT      ->CONTINUE
8E9C- 68        3720 AUTOCR   PLA             RESTORE STACK
8E9D- D0 29     3730          BNE END2        ->EXIT
8E9F- 68        3740 FULL     PLA             RESTORE STACK
8EA0- 20 3A FF  3750          JSR BELL        SOUND WARNING
8EA3- A4 24     3760 END1     LDY HPOS        THIS GETS THE
8EA5- B1 28     3770          LDA (BASL),Y    KEYBOARD BUFFER
8EA7- 20 ED FD  3780          JSR COUT        AND DOS TO
8EAA- A9 88     3790          LDA #$88        WORK CORRECTLY
8EAC- E8        3800          INX
8EAD- 60        3810          RTS             ->EXIT
                3820 *
                3830 * PROGRAM KEYS
                3840 *
8EAE- 20 8B FD  3850 PGM      JSR CROUT1      <CR> WITH CLEAR
8EB1- A0 00     3860          LDY #00
8EB3- 8C 0F 91  3870          STY AUTFLG      TURN OFF AUTO LN
8EB6- B9 14 91  3880 STEP11   LDA PKTBP,Y     PRINT "PRESS KEY
8EB9- 20 ED FD  3890          JSR COUT        TO BE PROGRAMMED"
8EBC- C8        3900          INY
8EBD- C0 1B     3910          CPY #27
8EBF- D0 F5     3920          BNE STEP11
                3930 *
8EC1- 20 C4 90  3940          JSR KEY         GET PROGRAM KEY
8EC4- C9 9B     3950          CMP #$9B        ESC?
8EC6- D0 03     3960          BNE STEP12      ->NO
8EC8- A9 8D     3970 END2     LDA #$8D        <CR>
8ECA- 60        3980          RTS             ->EXIT
8ECB- 20 B8 90  3990 STEP12   JSR PRTCHR      PRINT IT
8ECE- 20 8B 90  4000          JSR IJKMCK      ILLEGAL KEY CHECK
8ED1- B0 14     4010          BCS ILLGL       ->I,J,K, OR M
8ED3- C9 80     4020          CMP #$80        CTRL-SHIFT-P?
8ED5- F0 10     4030          BEQ ILLGL       ->YES
8ED7- C9 90     4040          CMP #$90        CTRL-P?
8ED9- F0 0C     4050          BEQ ILLGL       ->YES
8EDB- C9 84     4060          CMP #$84        CTRL-D?
8EDD- F0 08     4070          BEQ ILLGL       ->YES
8EDF- C9 8C     4080          CMP #$8C        CTRL-L?
8EE1- F0 04     4090          BEQ ILLGL       ->YES
8EE3- C9 9A     4100          CMP #$9A        CTRL-Z?
8EE5- D0 12     4110          BNE PGMCHK      ->NO
8EE7- 20 8B FD  4120 ILLGL    JSR CROUT1      <CR> WITH CLEAR
8EEA- A0 00     4130          LDY #00
8EEC- B9 47 91  4140 STEP13   LDA RK,Y        PRINT "RESERVED KEY"
8EEF- 20 ED FD  4150          JSR COUT
8EF2- C8        4160          INY
8EF3- C0 0D     4170          CPY #13
8EF5- D0 F5     4180          BNE STEP13
8EF7- F0 CF     4190          BEQ END2        ->EXIT
                4200 *
8EF9- 20 9B 90  4210 PGMCHK   JSR FIND        IS KEY PROGRAMMED?
8EFC- 90 12     4220          BCC LEGAL       ->NO
8EFE- 20 8B FD  4230          JSR CROUT1      <CR> WITH CLEAR
8F01- A0 00     4240          LDY #00
8F03- B9 54 91  4250 STEP14   LDA KIP,Y       PRINT "KEY IS
8F06- 20 ED FD  4260          JSR COUT        PROGRAMMED"
8F09- C8        4270          INY
8F0A- C0 12     4280          CPY #18
8F0C- D0 F5     4290          BNE STEP14
8F0E- F0 B8     4300          BEQ END2        ->EXIT
                4310 *
8F10- A5 18     4320 LEGAL    LDA TEMP        LEGAL KEY
8F12- 49 80     4330          EOR #$80        CLEAR BIT7
8F14- 20 D3 90  4340          JSR STOR1       STORE IT
8F17- 20 8B FD  4350          JSR CROUT1      <CR> WITH CLEAR
8F1A- A0 00     4360          LDY #00
8F1C- B9 85 91  4370 STEP15   LDA IPD,Y       PRINT "INPUT
8F1F- 20 ED FD  4380          JSR COUT        PROGRAM DATA"
8F22- C8        4390          INY
8F23- C0 13     4400          CPY #19
8F25- D0 F5     4410          BNE STEP15
                4420 *
8F27- 20 C4 90  4430 NXCHR    JSR KEY         GET DATA
8F2A- C9 95     4440          CMP #$95        FORWARD ARROW?
8F2C- F0 F9     4450          BEQ NXCHR       YES->IGNORE IT
8F2E- C9 88     4460          CMP #$88        BACK ARROW?
8F30- F0 10     4470          BEQ BCKARR      ->YES
8F32- C9 8D     4480          CMP #$8D        END?
8F34- F0 3C     4490          BEQ SAVE        ->YES
8F36- C9 92     4500          CMP #$92        CTRL-R?
8F38- F0 35     4510          BEQ AUTCR       ->YES
8F3A- 20 B8 90  4520          JSR PRTCHR      PRINT CHAR.
8F3D- 20 D3 90  4530          JSR STOR1       STORE CHAR.
8F40- D0 E5     4540          BNE NXCHR       ->NEXT CHARACTER
8F42- 20 ED FD  4550 BCKARR   JSR COUT        PRINT IT
8F45- 48        4560          PHA             SAVE IT
8F46- A5 06     4570          LDA STOR        DECREMENT THE
8F48- 38        4580          SEC             STORAGE POINTER
8F49- E9 01     4590          SBC #01
8F4B- 85 06     4600          STA STOR
8F4D- A5 07     4610          LDA STOR+1
8F4F- E9 00     4620          SBC #00
8F51- 85 07     4630          STA STOR+1
8F53- A0 00     4640          LDY #00
8F55- B1 06     4650          LDA (STOR),Y    BACKED UP TOO FAR?
8F57- 10 0E     4660          BPL TOOFAR      ->YES
8F59- A9 A0     4670          LDA #$A0        DELETE A CHAR. ON
8F5B- 20 ED FD  4680          JSR COUT        THE SCREEN
8F5E- 68        4690          PLA             RESTORE INPUT
8F5F- 20 ED FD  4700          JSR COUT
8F62- 98        4710          TYA             DELETE CHARACTER
8F63- 91 06     4720          STA (STOR),Y    IN THE BUFFER
8F65- F0 C0     4730          BEQ NXCHR       ->NEXT CHARACTER
8F67- 20 D7 90  4740 TOOFAR   JSR INC         INCREMENT STORAGE POINTER
8F6A- E6 24     4750          INC HPOS
8F6C- 68        4760          PLA             RESTORE INPUT
8F6D- D0 B8     4770          BNE NXCHR       ->NEXT CHARACTER
8F6F- 20 D3 90  4780 AUTCR    JSR STOR1       STORE CTRL-R
                4790 *
                4800 * SAVE TO DISK
                4810 *
8F72- 20 8B FD  4820 SAVE     JSR CROUT1      PRINT <CR>
8F75- A0 00     4830          LDY #00
8F77- B9 D9 91  4840 STEP16   LDA STD,Y       PRINT "SAVE
8F7A- 20 ED FD  4850          JSR COUT        TO DISK?"
8F7D- C8        4860          INY
8F7E- C0 13     4870          CPY #19
8F80- D0 F5     4880          BNE STEP16
                4890 *
8F82- 20 C4 90  4900          JSR KEY         GET ANSWER
8F85- C9 D9     4910          CMP #$D9        Y?
8F87- D0 17     4920          BNE END3        NO->EXIT
8F89- A5 06     4930          LDA STOR        SAVE STORAGE POINTER
8F8B- 8D FE 95  4940          STA TEMSTO
8F8E- A5 07     4950          LDA STOR+1
8F90- 8D FF 95  4960          STA TEMSTO+1
8F93- A2 00     4970          LDX #00
8F95- BD BE 91  4980 TRNSFR   LDA BSAVE,X     SAVE DATA
8F98- 9D 00 02  4990          STA KEYBRD,X    TO DISK
8F9B- E8        5000          INX
8F9C- E0 1B     5010          CPX #27
8F9E- D0 F5     5020          BNE TRNSFR
8FA0- A9 8D     5030 END3     LDA #$8D        <CR>
8FA2- 60        5040          RTS
                5050 *
                5060 * DELETE A PROGRAMMED KEY
                5070 *
8FA3- 20 8B FD  5080 DELETE   JSR CROUT1      <CR> WITH CLEAR
8FA6- A0 00     5090          LDY #00
8FA8- 8C 0F 91  5100          STY AUTFLG      TURN OFF AUTO LN
8FAB- B9 2F 91  5110 STEP17   LDA PKTBC,Y     PRINT "PRESS KEY
8FAE- 20 ED FD  5120          JSR COUT        TO BE CLEARED"
8FB1- C8        5130          INY
8FB2- C0 18     5140          CPY #24
8FB4- D0 F5     5150          BNE STEP17
                5160 *
8FB6- 20 C4 90  5170          JSR KEY         GET KEY TO CLEAR
8FB9- C9 9B     5180          CMP #$9B        ESC?
8FBB- F0 E3     5190          BEQ END3        YES->EXIT
8FBD- 20 B8 90  5200          JSR PRTCHR      PRINT CHAR.
8FC0- 20 9B 90  5210          JSR FIND        IS KEY PROGRAMMED?
8FC3- 90 34     5220          BCC NOTPGM      ->NO
8FC5- A5 08     5230          LDA READ        SET STOR TO
8FC7- 85 06     5240          STA STOR        BEGINNING ADDRESS OF
8FC9- A5 09     5250          LDA READ+1      THE PROGRAMMED KEY
8FCB- 85 07     5260          STA STOR+1      TO DELETE
8FCD- A0 00     5270          LDY #00
8FCF- 20 FF 90  5280 STEP18   JSR INCR        INCREMENT READ POINTER
8FD2- B1 08     5290          LDA (READ),Y    SET READ TO END OF
8FD4- 30 F9     5300          BMI STEP18      PROGRAMMED KEY+1
8FD6- B1 08     5310 STEP19   LDA (READ),Y    MOVE DATA DOWN
8FD8- F0 08     5320          BEQ LAST        OVER THE DELETED AREA
8FDA- 20 D3 90  5330          JSR STOR1
8FDD- 20 FF 90  5340          JSR INCR
8FE0- D0 F4     5350          BNE STEP19
                5360 *
```

```
8FE2- A5 06    5370 LAST     LDA STOR        DATA MOVED DOWN,
8FE4- 85 08    5380          STA READ        NOW ZERO REMAINING
8FE6- A5 07    5390          LDA STOR+1      DATA
8FE8- 85 09    5400          STA READ+1
8FEA- A0 00    5410          LDY #00
8FEC- B1 08    5420 ZERO     LDA (READ),Y
8FEE- F0 82    5430          BEQ SAVE        ->FINISHED
8FF0- A9 00    5440          LDA #00
8FF2- 91 08    5450          STA (READ),Y
8FF4- 20 FF 90 5460          JSR INCR        INCREMENT READ POINTER
8FF7- D0 F3    5470          BNE ZERO
               5480 *
8FF9- 20 8B FD 5490 NOTPGM   JSR CROUT1      <CR> WITH CLEAR
8FFC- A0 00    5500          LDY #00
8FFE- B9 66 91 5510 STEP20   LDA KNP,Y       PRINT "KEY NOT
9001- 20 ED FD 5520          JSR COUT        PROGRAMMED"
9004- C8       5530          INY
9005- C0 13    5540          CPY #19
9007- D0 F5    5550          BNE STEP20
9009- F0 95    5560          BEQ END3        ->EXIT
               5570 *
               5580 * LIST PROGRAMMED KEYS
               5590 *
900B- A9 92    5600 LIST     LDA /BUFFER     SET READ POINTER
900D- 85 09    5610          STA READ+1
900F- A9 00    5620          LDA #BUFFER
9011- 85 08    5630          STA READ
9013- 20 58 FC 5640          JSR HOME        CLEAR SCREEN
9016- A0 00    5650 STEP21   LDY #00
9018- B1 08    5660          LDA (READ),Y    GET CHARACTER
901A- F0 3C    5670          BEQ FRSPC       ->FINISHED
901C- 30 32    5680          BMI CHR         ->DATA
901E- 48       5690          PHA
901F- 20 8B FD 5700          JSR CROUT1      <CR> WITH CLEAR
9022- A5 25    5710          LDA VPOS        SCREEN FULL?
9024- C9 12    5720          CMP #18
9026- 90 16    5730          BCC STEP23      NO->CONTINUE
9028- 20 8B FD 5740          JSR CROUT1      <CR> WITH CLEAR
902B- A0 00    5750          LDY #00
902D- B9 98 91 5760 STEP22   LDA PAKTC,Y     PRINT "PRESS ANY
9030- 20 ED FD 5770          JSR COUT        KEY TO CONTINUE"
9033- C8       5780          INY
9034- C0 1B    5790          CPY #27
9036- D0 F5    5800          BNE STEP22
               5810 *
9038- 20 1B FD 5820          JSR KEYIN       WAIT FOR KEYPRESS
903B- 20 58 FC 5830          JSR HOME        CLEAR SCREEN
903E- 68       5840 STEP23   PLA
903F- 49 80    5850          EOR #$80
9041- 20 B8 90 5860          JSR PRTCHR      PRINT PROGRAM KEY
9044- A9 A0    5870          LDA #$A0        SPACE
9046- 20 ED FD 5880          JSR COUT        PRINT IT
9049- A9 AD    5890          LDA #$AD        "-"
904B- 20 ED FD 5900          JSR COUT        PRINT IT
904E- A9 A0    5910          LDA #$A0        SPACE
9050- 20 B8 90 5920 CHR      JSR PRTCHR      PRINT IT
9053- 20 FF 90 5930          JSR INCR        INCREMENT READ POINTER
9056- D0 BE    5940          BNE STEP21      ->NEXT
               5950 *
9058- 20 8B FD 5960 FRSPC    JSR CROUT1      <CR> WITH CLEAR
905B- 20 8B FD 5970          JSR CROUT1
905E- A9 FD    5980          LDA #BUFEND-3   PRINT FREE
9060- 38       5990          SEC             BUFFER SPACE
9061- E5 06    6000          SBC STOR
9063- AA       6010          TAX
9064- A9 95    6020          LDA /BUFEND-1
9066- E5 07    6030          SBC STOR+1
9068- AC 01 E0 6040          LDY INTCK       INTEGER?
906B- F0 06    6050          BEQ INTGR       ->YES
906D- 20 24 ED 6060          JSR LINPRT
9070- 4C 76 90 6070          JMP STEP24
9073- 20 1B E5 6080 INTGR    JSR INTPRT
9076- A0 00    6090 STEP24   LDY #00
9078- 8C 0F 91 6100          STY AUTFLG      TURN OFF AUTO LN
907B- B9 B3 91 6110 STEP25   LDA BYFR,Y      PRINT "BYTES FREE"
907E- 20 ED FD 6120          JSR COUT
9081- C8       6130          INY
9082- C0 0B    6140          CPY #11
9084- D0 F5    6150          BNE STEP25
9086- A2 00    6160 END4     LDX #00
9088- A9 8D    6170          LDA #$8D        <CR>
908A- 60       6180          RTS             ->EXIT
               6190 *
               6200 * MISC. SUBROUTINES
               6210 *
908B- C9 CE    6220 IJKMCK   CMP #$CE        N?
908D- B0 0A    6230          BCS CLEAR       ->N OR GREATER
908F- C9 C9    6240          CMP #$C9        I?
9091- 90 07    6250          BCC RTS1        ->LESS THAN I
9093- C9 CC    6260          CMP #$CC        L?
9095- F0 02    6270          BEQ CLEAR       ->YES
9097- 38       6280          SEC             I,J,K, OR M
9098- 60       6290          RTS
9099- 18       6300 CLEAR    CLC             NOT I,J,K, OR M
909A- 60       6310 RTS1     RTS
               6320 *
909B- 85 18    6330 FIND     STA TEMP        STORE INPUT
909D- A9 91    6340          LDA /BUFFER-1   SET READ POINTER
909F- 85 09    6350          STA READ+1
90A1- A9 FF    6360          LDA #BUFFER-1
90A3- 85 08    6370          STA READ
90A5- A0 00    6380          LDY #00
90A7- 20 FF 90 6390 STEP26   JSR INCR        INCREMENT READ POINTER
90AA- B1 08    6400          LDA (READ),Y
90AC- F0 EB    6410          BEQ CLEAR       ->KEY NOT PROGRAMMED
90AE- 30 F7    6420          BMI STEP26      ->DATA, TRY AGAIN
90B0- 49 80    6430          EOR #$80
90B2- C5 18    6440          CMP TEMP
90B4- D0 F1    6450          BNE STEP26      ->WRONG KEY
90B6- 38       6460          SEC             KEY IS PROGRAMMED
90B7- 60       6470          RTS
               6480 *
90B8- C9 9F    6490 PRTCHR   CMP #$9F        CTRL-CHARACTER?
90BA- B0 02    6500          BCS STEP27      ->NO
90BC- 49 80    6510          EOR #$80        MAKE IT PRINTABLE
90BE- 20 ED FD 6520 STEP27   JSR COUT        PRINT CHARACTER
90C1- 09 80    6530          ORA #$80
90C3- 60       6540          RTS
               6550 *
90C4- A4 24    6560 KEY      LDY HPOS        KEYBOARD INPUT
90C6- B1 28    6570          LDA (BASL),Y    FLASH THE CURSOR
90C8- 48       6580          PHA
90C9- 29 3F    6590          AND #$3F
90CB- 09 40    6600          ORA #$40
90CD- 91 28    6610          STA (BASL),Y
90CF- 68       6620          PLA
90D0- 4C 1B FD 6630          JMP KEYIN       GET INPUT
               6640 *
90D3- A0 00    6650 STOR1    LDY #00         STORE DATA IN
90D5- 91 06    6660          STA (STOR),Y    PROGRAMS BUFFER
90D7- E6 06    6670 INC      INC STOR        INCREMENT STORAGE
90D9- F0 21    6680          BEQ STEP29      POINTER
90DB- A5 06    6690          LDA STOR
90DD- C9 FD    6700          CMP #BUFEND-3   END OF BUFFER?
90DF- D0 1D    6710          BNE RTS2        ->NO
90E1- A5 07    6720          LDA STOR+1
90E3- C9 95    6730          CMP /BUFEND-1
90E5- D0 17    6740          BNE RTS2        ->NO
90E7- 20 8B FD 6750          JSR CROUT1
90EA- A0 00    6760          LDY #00
90EC- B9 79 91 6770 STEP28   LDA BF,Y        PRINT "BUFFER FULL"
90EF- 20 ED FD 6780          JSR COUT
90F2- C8       6790          INY
90F3- C0 0C    6800          CPY #12
90F5- D0 F5    6810          BNE STEP28
90F7- 68       6820          PLA             RESET STACK
90F8- 68       6830          PLA
90F9- A9 8D    6840          LDA #$8D        <CR>
90FB- 60       6850          RTS
90FC- E6 07    6860 STEP29   INC STOR+1
90FE- 60       6870 RTS2     RTS
               6880 *
90FF- E6 08    6890 INCR     INC READ        INCREMENT READ
9101- D0 02    6900          BNE RTS3        POINTER
9103- E6 09    6910          INC READ+1
9105- 60       6920 RTS3     RTS
               6930 *
9106- A9 92    6940 SET      LDA /BUFFER     SET STORAGE POINTER
9108- 85 07    6950          STA STOR+1
910A- A9 00    6960          LDA #BUFFER
910C- 85 06    6970          STA STOR
910E- 60       6980          RTS
               6990 *
               7000 * TABLES & FLAG
               7010 *
910F- 00       7020 AUTFLG   .DA #00          AUTO LN FLAG
9110- C1 D5 D4
9113- CF       7030 AUTO     .AS -"AUTO"
9114- D0 D2 C5
9117- D3 D3 A0
911A- CB C5 D9
911D- A0 D4 CF
9120- A0 C2 C5
9123- A0 D0 D2
9126- CF C7 D2
9129- C1 CD CD
912C- C5 C4 BA 7040 PKTBP    .AS -"PRESS KEY TO BE PROGRAMMED:"
912F- D0 D2 C5
9132- D3 D3 A0
9135- CB C5 D9
9138- A0 D4 CF
913B- A0 C2 C5
913E- A0 C3 CC
9141- C5 C1 D2
9144- C5 C4 BA 7050 PKTBC    .AS -"PRESS KEY TO BE CLEARED:"
9147- D2 C5 D3
914A- C5 D2 D6
914D- C5 C4 A0
9150- CB C5 D9 7060 RK       .AS -"RESERVED KEY"
9153- 87       7070          .HS 87
9154- CB C5 D9
9157- A0 C9 D3
915A- A0 D0 D2
915D- CF C7 D2
9160- C1 CD CD
9163- C5 C4    7080 KIP      .AS -"KEY IS PROGRAMMED"
9165- 87       7090          .HS 87
9166- CB C5 D9
9169- A0 CE CF
916C- D4 A0 D0
916F- D2 CF C7
9172- D2 C1 CD
9175- CD C5 C4 7100 KNP      .AS -"KEY NOT PROGRAMMED"
9178- 87       7110          .HS 87
9179- C2 D5 C6
917C- C6 C5 D2
917F- A0 C6 D5
9182- CC CC    7120 BF       .AS -"BUFFER FULL"
9184- 87       7130          .HS 87
9185- C9 CE D0
9188- D5 D4 A0
918B- D0 D2 CF
918E- C7 D2 C1
9191- CD A0 C4
9194- C1 D4 C1
9197- BA       7140 IPD      .AS -"INPUT PROGRAM DATA:"
9198- BC D0 D2
919B- C5 D3 D3
919E- A0 C1 CE
91A1- D9 A0 CB
91A4- C5 D9 A0
91A7- D4 CF A0
91AA- C3 CF CE
91AD- D4 C9 CE
91B0- D5 C5 BE 7150 PAKTC    .AS -"<PRESS ANY KEY TO CONTINUE>"
91B3- A0 C2 D9
91B6- D4 C5 D3
91B9- A0 C6 D2
91BC- C5 C5    7160 BYFR     .AS -" BYTES FREE"
91BE- C2 D3 C1
91C1- D6 C5 A0
91C4- D0 C7 CD
91C7- AE C4 C1
91CA- D4 C1 AC
91CD- C1 A4 B9
91D0- B2 B0 B0
91D3- AC CC A4
91D6- B4 B0 B0 7170 BSAVE    .AS -"BSAVE PGM.DATA,A$9200,L$400"
```

```
91D9- D3 C1 D6
91DC- C5 A0 D4
91DF- CF A0 C4
91E2- C9 D3 CB
91E5- BF A0 A8
91E8- D9 AF CE
91EB- A9        7180 STD        .AS -"SAVE TO DISK? (Y/N)"
91EC- C2 CC CF
91EF- C1 C4 A0
91F2- D0 C7 CD
91F5- AE C4 C1
91F8- D4 C1     7190 BPD        .AS -"BLOAD PGM.DATA"
91FA- C4 C2 C1
91FD- FF C3     7200 TBL        .HS C4C2C1FFC3
                7210            .EN
```

END OF LISTING 1

```
        KEY PERFECT 4.0
            RUN ON
         POWER.KEY
=============================
   CODE       ADDR# - ADDR#
  ------      -------------
    2A03       8CF0 - 8D3F
    29C9       8D40 - 8D8F
    26E5       8D90 - 8DDF
    24AB       8DE0 - 8E2F
    229E       8E30 - 8E7F
    269F       8E80 - 8ECF
    22E3       8ED0 - 8F1F
    2843       8F20 - 8F6F
    28F9       8F70 - 8FBF
    280B       8FC0 - 900F
    2322       9010 - 905F
    243D       9060 - 90AF
    2638       90B0 - 90FF
```

```
    23CD       9100 - 914F
    29B2       9150 - 919F
    27B5       91A0 - 91EF
    0530       91F0 - 91FD
PROGRAM CHECK IS : 050E
```

```
CHECK CODE 3.0

ON: POWER.KEY

TYPE: B

LENGTH: 050E
CHECKSUM: 08
```

LISTING 2: PGM.DATA

```
9200- 42 C2 D3 C1 D6 C5 02 C2
9208- CC CF C1 C4 43 C3 C1 D4
9210- C1 CC CF C7 AC C4 B1 92
9218- 03 C3 C1 D4 C1 CC CF C7
9220- AC C4 B2 92 44 C4 C5 CC
9228- C5 D4 C5 46 C6 CF D2 A0
9230- C9 BD 06 C6 CC C1 D3 C8
9238- 47 C7 CF D4 CF 07 C7 CF
9240- D3 D5 C2 48 C8 CF CD C5
9248- 08 C8 D4 C1 C2 09 C9 CE
9250- D0 D5 D4 0B C3 C8 D2 A4
9258- A8 4C CC C9 D3 D4 92 0D
9260- CD C9 C4 A4 A8 4E CE C5
9268- D8 D4 0E CE CF D2 CD C1
9270- CC 50 D0 D2 C9 CE D4 52
9278- D2 C5 D4 D5 D2 CE 12 D2
9280- C9 C7 C8 D4 A4 A8 53 D3
9288- C1 D6 C5 13 D3 D4 D2 A4
9290- A8 54 D4 C5 D8 D4 14 D4
9298- C8 C5 CE 56 D6 D4 C1 C2
92A0- 3C CC C5 C6 D4 A4 A8 5E
92A8- C9 CE D6 C5 D2 D3 C5 00
92B0- 00 00 00 00 00 00 00 00
92B8- 00 00 00 00 00 00 00 00
92C0- 00 00 00 00 00 00 00 00
92C8- 00 00 00 00 00 00 00 00
92D0- 00 00 00 00 00 00 00 00
92D8- 00 00 00 00 00 00 00 00
92E0- 00 00 00 00 00 00 00 00
92E8- 00 00 00 00 00 00 00 00
92F0- 00 00 00 00 00 00 00 00
92F8- 00 00 00 00 00 00 00 00
9300- 00 00 00 00 00 00 00 00
9308- 00 00 00 00 00 00 00 00
9310- 00 00 00 00 00 00 00 00
9318- 00 00 00 00 00 00 00 00
9320- 00 00 00 00 00 00 00 00
9328- 00 00 00 00 00 00 00 00
9330- 00 00 00 00 00 00 00 00
9338- 00 00 00 00 00 00 00 00
9340- 00 00 00 00 00 00 00 00
9348- 00 00 00 00 00 00 00 00
9350- 00 00 00 00 00 00 00 00
9358- 00 00 00 00 00 00 00 00
9360- 00 00 00 00 00 00 00 00
9368- 00 00 00 00 00 00 00 00
9370- 00 00 00 00 00 00 00 00
9378- 00 00 00 00 00 00 00 00
9380- 00 00 00 00 00 00 00 00
9388- 00 00 00 00 00 00 00 00
9390- 00 00 00 00 00 00 00 00
9398- 00 00 00 00 00 00 00 00
93A0- 00 00 00 00 00 00 00 00
93A8- 00 00 00 00 00 00 00 00
93B0- 00 00 00 00 00 00 00 00
93B8- 00 00 00 00 00 00 00 00
93C0- 00 00 00 00 00 00 00 00
93C8- 00 00 00 00 00 00 00 00
93D0- 00 00 00 00 00 00 00 00
93D8- 00 00 00 00 00 00 00 00
93E0- 00 00 00 00 00 00 00 00
93E8- 00 00 00 00 00 00 00 00
93F0- 00 00 00 00 00 00 00 00
93F8- 00 00 00 00 00 00 00 00
9400- 00 00 00 00 00 00 00 00
9408- 00 00 00 00 00 00 00 00
9410- 00 00 00 00 00 00 00 00
9418- 00 00 00 00 00 00 00 00
9420- 00 00 00 00 00 00 00 00
9428- 00 00 00 00 00 00 00 00
9430- 00 00 00 00 00 00 00 00
9438- 00 00 00 00 00 00 00 00
9440- 00 00 00 00 00 00 00 00
9448- 00 00 00 00 00 00 00 00
9450- 00 00 00 00 00 00 00 00
9458- 00 00 00 00 00 00 00 00
9460- 00 00 00 00 00 00 00 00
9468- 00 00 00 00 00 00 00 00
9470- 00 00 00 00 00 00 00 00
9478- 00 00 00 00 00 00 00 00
9480- 00 00 00 00 00 00 00 00
9488- 00 00 00 00 00 00 00 00
9490- 00 00 00 00 00 00 00 00
9498- 00 00 00 00 00 00 00 00
94A0- 00 00 00 00 00 00 00 00
94A8- 00 00 00 00 00 00 00 00
94B0- 00 00 00 00 00 00 00 00
94B8- 00 00 00 00 00 00 00 00
94C0- 00 00 00 00 00 00 00 00
94C8- 00 00 00 00 00 00 00 00
94D0- 00 00 00 00 00 00 00 00
94D8- 00 00 00 00 00 00 00 00
94E0- 00 00 00 00 00 00 00 00
94E8- 00 00 00 00 00 00 00 00
94F0- 00 00 00 00 00 00 00 00
94F8- 00 00 00 00 00 00 00 00
9500- 00 00 00 00 00 00 00 00
9508- 00 00 00 00 00 00 00 00
9510- 00 00 00 00 00 00 00 00
9518- 00 00 00 00 00 00 00 00
9520- 00 00 00 00 00 00 00 00
9528- 00 00 00 00 00 00 00 00
9530- 00 00 00 00 00 00 00 00
9538- 00 00 00 00 00 00 00 00
9540- 00 00 00 00 00 00 00 00
9548- 00 00 00 00 00 00 00 00
9550- 00 00 00 00 00 00 00 00
9558- 00 00 00 00 00 00 00 00
9560- 00 00 00 00 00 00 00 00
9568- 00 00 00 00 00 00 00 00
9570- 00 00 00 00 00 00 00 00
9578- 00 00 00 00 00 00 00 00
9580- 00 00 00 00 00 00 00 00
9588- 00 00 00 00 00 00 00 00
9590- 00 00 00 00 00 00 00 00
9598- 00 00 00 00 00 00 00 00
95A0- 00 00 00 00 00 00 00 00
95A8- 00 00 00 00 00 00 00 00
95B0- 00 00 00 00 00 00 00 00
95B8- 00 00 00 00 00 00 00 00
95C0- 00 00 00 00 00 00 00 00
95C8- 00 00 00 00 00 00 00 00
95D0- 00 00 00 00 00 00 00 00
95D8- 00 00 00 00 00 00 00 00
95E0- 00 00 00 00 00 00 00 00
95E8- 00 00 00 00 00 00 00 00
95F0- 00 00 00 00 00 00 00 00
95F8- 00 00 00 00 00 00 AF 92
```

END OF LISTING 2

```
        KEY PERFECT 4.0
            RUN ON
          PGM.DATA
=============================
   CODE       ADDR# - ADDR#
  ------      -------------
    2370       9200 - 924F
    2645       9250 - 929F
    28F9       92A0 - 92EF
      00       92F0 - 933F
      00       9340 - 938F
      00       9390 - 93DF
      00       93E0 - 942F
      00       9430 - 947F
      00       9480 - 94CF
      00       94D0 - 951F
      00       9520 - 956F
      00       9570 - 95BF
      FA       95C0 - 95FF
PROGRAM CHECK IS : 0400
```

```
CHECK CODE 3.0
ON: PGM.DATA
TYPE: B
LENGTH: 0400
CHECKSUM: 97
```