

**DAKIN5**  
CORPORATION

**PROGRAMMING AIDS 3.3**

**DAKIN5**  
CORPORATION  
**PROGRAMMING AIDS**

**DAKIN5**  
CORPORATION  
**PROGRAMMING AIDS 3.3**

**DAKIN5**  
CORPORATION

**Programming  
Aids  
3.3**

**12**

**12-in-1 Utility Kit**

## NOTICE

Dakin5 Corporation reserves the right to make improvements in the product described in this manual at any time and without notice.

## DISCLAIMER OF ALL WARRANTIES AND LIABILITY

DAKIN5 CORPORATION MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. DAKIN5 CORPORATION SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT DAKIN5 CORPORATION, ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL DAKIN5 CORPORATION BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF DAKIN5 CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted and contains proprietary information. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Dakin5 Corporation.

Willful violations of the Copyright Law of the United States can result in civil damages of up to \$50,000 in addition to actual damages, plus criminal penalties of up to one year imprisonment and/or a \$10,000 fine.

© 1980 by DAKIN5 CORPORATION  
Post Office Box 21187  
Denver, Colorado 80221  
(303) 426-6090

All rights reserved.

APPLE COMPUTER INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

## NOTICE

Dakin5 Corporation reserves the right to make improvements in the product described in this manual at any time and without notice.

## DISCLAIMER OF ALL WARRANTIES AND LIABILITY

DAKIN5 CORPORATION MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. DAKIN5 CORPORATION SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT DAKIN5 CORPORATION, ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL DAKIN5 CORPORATION BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF DAKIN5 CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted and contains proprietary information. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Dakin5 Corporation.

Willful violations of the Copyright Law of the United States can result in civil damages of up to \$50,000 in addition to actual damages, plus criminal penalties of up to one year imprisonment and/or a \$10,000 fine.

© 1980 by DAKIN5 CORPORATION  
Post Office Box 21187  
Denver, Colorado 80221  
(303) 426-6090

All rights reserved.

APPLE COMPUTER INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.



## INTRODUCTION

Dakin5 Corporation, a major software house, is making available to the public 12 utility programs on one 16 sector diskette, utilizing the new Apple\* DOS 3.3.

The Dakin5 Programming Aids 3.3 have been developed and tested for in-house use while producing business application software, such as The Controller<sup>TM</sup>\*\* for Apple Computer Inc.

These utilities will facilitate the development of your own microcomputer programs.

You are advised to choose selection 7, THE DISKETTE COPY, and make an immediate backup copy of your Dakin5 Programming Aids 3.3 diskette.

The hardware needed to run these DOS 3.3 utilities is an Apple II computer with Applesoft, one disk drive (except for the Copier and the Diskette Copy which require two drives), and a video monitor or television set. A printer (compatible with your Apple II computer) is necessary for the Screen Printer and the Lister. A printer is optional for the Line Cross Reference, the Variable Cross Reference, the Peeker and the Array Editor.

All of the Dakin5 Programming Aids 3.3 programs are compatible with the Corvus Disk Drive system.

In order to run this Dakin5 Programming Aids 3.3 package, you must have the DOS 3.3 disk controller chips; your program diskette contains the 3.3 DOS.

The Screen Printer, Peeker, and Prompter all use 16K of memory if Applesoft is in ROM. If not, they require 24K. All other programs require 48K.

The Dakin5 Programming Aids 3.3 package includes the following menu-driven programs:

1. LISTER
2. LINE REFERENCE
3. VARIABLE REFERENCE
4. PEEKER
5. PATCHER
6. COPIER
7. DISKETTE COPY
8. ARRAY EDITOR
9. CALCULATOR
10. SCREEN PRINTER
11. PROMPTER
12. CRUNCHER

### LISTER

Sends BASIC programs to the printer to be listed, utilizing the full line capacity of the printer. Pagination and page headings, including program name and date, are also provided as additional options.

\*Apple is a registered trademark of Apple Computer Inc.

\*\*The Controller is a registered trademark of Dakin5 Corporation.

## LINE CROSS REFERENCE

Displays or prints a listing of all lines referenced by GOTO, THEN, GOSUB, LIST or RUN statements in an Applesoft BASIC program. Cross-referencing of most programs is done in a few seconds. An option allows you to print only the line numbers referenced in GOSUB statements.

## VARIABLE CROSS REFERENCE

Displays or prints a listing of all variable names used in an Applesoft BASIC program, showing all line numbers where a given variable name is used.

## PEEKER

Displays or prints either all or selected records from a text file.

## PATCHER

Allows you to display any sector of a given file or program, and then to update any data within that sector. Another option permits you to specify the sector you wish to update, such as directory sectors and sectors occupied by DOS.

## COPIER

Copies any type of file or program on a normally formatted diskette from one diskette to another. The name of the program or file is the only information needed

## DISKETTE COPY

Is a diskette-to-diskette copy program that does more than just copy. First, the program verifies the input. Then it formats an output disk, copies each track, and checks that the output matches the input. Additional options allow you to either initialize a diskette without DOS, or to create a copy without DOS, thereby increasing storage by 32 sectors. You may even create a copy with a different volume number than the original. Use this Diskette Copy immediately to make a backup copy of your Dakin5 Programming Aids 3.3 diskette.

## ARRAY EDITOR

Is a simple word processor that allows you to create, modify, print, and save your own text or EXEC files.

## CALCULATOR

Adds, subtracts, multiplies and divides very large numbers using numeric string data. The Calculator subroutine (using twenty place accuracy) is written in Assembler code, and runs much faster than an equivalent BASIC subroutine.



## SCREEN PRINTER

Permits contents of the text screen to be sent to the printer at any time the keyboard is active (i.e., the cursor is visible). This Screen Printer program remains in effect until you press RESET or "reboot" the system.

## PROMPTER

Is a data entry subroutine that handles both string and numeric data. You have the option of using commas, decimal points and leading zeros with right-justified numerics. Alphanumeric data is left justified with trailing spaces added as required. With the Prompter you are also able to specify maximum field length to prevent overflow in both numeric and alphanumeric fields. You can even define your own set of valid characters.

## CRUNCHER

Compresses code in Applesoft programs. It also removes unreferenced (dead) code, unreferenced REM (remark) statements, and the comment after the REM in referenced REM statements. This will increase the speed of your programs; memory and disk space savings could be more than 45%.

## SPECIAL KEY FUNCTIONS

### THE ESCAPE KEY

The ESCape key returns you to the previously displayed screen. The exceptions for the Dakin5 Programming Aids 3.3 are: 1) during the actual printing process of the Screen Printer and the Lister; 2) while the Copier and Diskette Copy programs are doing a copy; and 3) during the running of the Cruncher.

If you press the ESCape key while at the DAKIN5 PROGRAMMING AIDS 3.3 MENU, your Apple exits to Applesoft and you are shown:

```
      DAKIN5 PROGRAMMING AIDS 3.3 COMPLETED.  
      ]
```

### THE RESET KEY

Anytime you press the RESET key while a programming aid is actually processing information, you should consider that particular output as invalid. You must run the program again to insure reliable results.

In the event that you press RESET during the Calculator, the Diskette Copy, the Copier, the Line Cross Reference, the Variable Cross Reference, the Patcher, or the Cruncher, you must reboot DOS. This is because these programs overlay portions of page 0 memory, and page 0 is not restored until the programs return to the PROGRAMMING AIDS 3.3 MENU.

During the printing of your program by the Lister, there is a situation that arises concerning the RESET key and the EXEC file. This is explained in detail in the section of the documentation dealing with the Lister.

## THE RESTORE KEY

The Restore or left arrow key < - either erases the previous character or, if a default exists, it replaces that character with the corresponding character in the default.

## THE RETURN KEY

The RETURN key, when pressed, performs its normal function as well as permitting you to accept default values as though you had entered the data yourself.

All the utility programs use the data entry routine demonstrated in Selection 11, THE PROMPTER. The following information, relevant to this routine, should be understood before continuing since it affects all the selections:

1. Whenever a default value is displayed, press RETURN, and the entire value will be accepted as though you had entered the data yourself.
2. If you are entering numeric data, you need only type the desired numeric characters. Do not include decimal points, commas or leading zeros.

## CORVUS DISK DRIVE USE

Whenever you are asked to indicate the slot and drive number you are using, you will also be instructed to enter VOLUME. If you have a Corvus Disk Drive system that uses volume numbers to simulate floppy disks, or if you use volume numbers on your diskette, enter the volume number. In all other cases, accept the default value which is zero (0).

## ADDITIONAL INFORMATION

The Dakin5 Programming Aids 3.3 diskette can be booted with either a Pascal Language card or a firmware APPLESOFT card. Users with Applesoft in RAM must delete the programs called APPLESOFT and FPBASIC, and replace them with your RAM version of APPLESOFT.

Pressing the ESCape key while at the menu returns you to Applesoft. The "HELLO" program is UT0002.

All selections are menu-driven. The date on the menu screen is changed by choosing Selection 13, CHANGE TODAY'S DATE. The numerics must be entered with appropriate leading zeros, combined with slashes between month and day, and between day and year, such as 09/10/79. This date on the menu is carried to the various programming aids as the default whenever a date is requested. This is the only date that remains in effect as you move from one aid to another, or turn off your Apple computer.

Do not write-protect your Dakin5 Programming Aids 3.3 diskette, or lock either the label file DLXX00 or the LISTER-EXEC file. Whenever you run the Lister program the LISTER-EXEC is updated; whenever you change TODAY'S DATE on the PROGRAMMING AIDS 3.3 MENU, the label file is updated. If you lock the label file, you will receive the error message DLXX00 IS LOCKED. You will then have to unlock it.

Every effort has been made to trap errors, and to provide error handling routines to ensure the proper running of this utility package. In the event that no appropriate error handling routine exists, you will receive the following error message on the screen:

```
ERROR XXX HAS OCCURRED. REF NO nnn
REFER TO YOUR USER'S MANUAL.
PRESS 'RETURN' KEY TO CONTINUE.
```

The xxx is the value of the Applesoft ONERR GOTO codes as explained in the Apple DOS 3.3 manual, or in the Applesoft II manual, page 136. The nnn represents the line number on which the error occurred. When you press RETURN, you will be returned to the DAKIN5 PROGRAMMING AIDS 3.3 MENU. This particular error message (and its explanation) is applicable to all the utility programs, but is not repeated in the ERROR MESSAGE sections of the Programming Aids 3.3 manual.

Whenever you boot your system from your programming aids diskette, you are shown the following menu:

```
** DAKIN5 PROGRAMMING AIDS 3.3 MENU **
*
*          TODAY'S DATE 05/19/80
*
*
*      1. THE LISTER
*      2. THE LINE REFERENCE
*      3. THE VARIABLE REFERENCE
*      4. THE PEEKER
*      5. THE PATCHER
*      6. THE COPIER
*      7. THE DISKETTE COPY
*      8. THE ARRAY EDITOR
*      9. THE CALCULATOR
*     10. THE SCREEN PRINTER
*     11. THE PROMPTER
*     12. THE CRUNCHER
*     13. CHANGE TODAY'S DATE
*
*
*
*          ENTER YOUR SELECTION -> ..
*
***** 1.0*****
```

Programs written by Rod Riggenbach

Documentation written by Allen Martin



LISTER

LISTER

WCEE  
Lease List Billing Products  
424 North Cedarbrook Avenue  
St. Charles, Missouri 63002-1417, 855-9-0822

## THE LISTER

The Lister enables you to print your BASIC programs utilizing the full line capacity of your printer. By selecting the pagination option, the Lister will number your pages and skip over the perforations. Using the Lister will ensure you of neat, easy-to-read BASIC program listings.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, 48K of memory, and 1 printer (compatible with the Apple). You may wish to modify line 8065 to be compatible with your printer. DOS must remain in effect at all times.

The Lister is reached by choosing Selection 1 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU.

The Lister screen shows one by one the following data entry prompts, with the dots indicating the maximum response length:

TODAY'S DATE (MM/DD/YY) .....

The default is the date as entered on the PROGRAMMING AIDS 3.3 MENU. You can, however, change the date on the Lister parameter screen. To correctly do so, the numerics must be entered with appropriate leading zeros, combined with slashes between month and day, and between day and year. Sample entries would be 09/10/79 and 12/03/78. The only date that remains in effect after you move from the Lister to another aid, or turn off your Apple, is the date shown on the PROGRAMMING AIDS 3.3 MENU.

TIME (HHMM) ....

The hours and minutes should be entered according to the 24 hour clock. For example, 2:30 p.m. should be entered as 1430. The default value is four zeros.

PROGRAM NAME

.....  
Enter the name of the program you wish to have listed.

SLOT .

Enter the number of the slot in which your disk controller card is located. The default is the slot of the last drive used.

DRIVE .

Enter the disk drive number to be used. The default is the number of the last drive used.

VOLUME ...

Accept the zero (0) default if you do not use volume numbers.

STARTING LINE .....

The default is 0. The permissible range for line numbers is from 0 to 63999.

ENDING LINE .....

The default is 63999. You may end at any desired line number.

NOTE: Line numbers must not exceed 32767 when listing Integer BASIC programs.

PRINTER SLOT .

This is the slot where your printer interface card is located. The default is 1.

PARALLEL INTERFACE CARD? .

Parallel interface card users should reply Y. If you are using a serial or communications interface card, reply N. The default value is Y.

PRINTER LINE LENGTH ...

The default value for printer line length is 80. The permissible range is from 40 to 255 columns.

DO YOU WANT PAGINATION? .

Accept the Y default to have your pages numbered (the Lister will skip six blank lines at the end of each page). You will then be prompted to indicate:

PRINTER PAGE LENGTH ..

The range is from 6 to 99, the default being 66.

You are next asked:

IS THIS ENTRY CORRECT? .

You may answer Y or N only. If there is an item that you want to change, answer N, and the cursor returns to the first item which is TODAY'S DATE. Your screen should look like:

\*\* THE LISTER \*\*

TODAY'S DATE (MM/DD/YY) 05/19/80

TIME (HHMM) 1430

PROGRAM NAME  
UT4800

SLOT 6 DRIVE 1 VOLUME 0

STARTING LINE 5 ENDING LINE 2055

PRINTER SLOT 1

PARALLEL INTERFACE CARD? Y

PRINTER LINE LENGTH 80

DO YOU WANT PAGINATION? Y

PRINTER PAGE LENGTH 66

IS THIS ENTRY CORRECT? .

It is a very simple and quick task to make changes since you need only to press RETURN to accept the default value for any items. In this case, the defaults represent the data (accepted as valid) you last entered.



When you are satisfied that all the information is correct, reply Y to the IS THIS ENTRY CORRECT? prompt, and the next screen will instruct you to:

TURN ON THE PRINTER AND ALIGN PAPER.  
PRESS 'RETURN' KEY TO CONTINUE.

Press RETURN, and the screen will prompt you to:

INSERT YOUR PROGRAM DISKETTE  
IN SLOT 6 DRIVE 1  
PRESS 'RETURN' KEY TO CONTINUE.

The red light for drive 1 will go on, and the drive will start spinning. Wait until the light goes off before inserting the program diskette, and pressing the RETURN key.

If you have two drives, and have indicated drive 2 when asked which drive (after entering the program name), the prompt will instruct you to insert the program diskette in drive 2. Press RETURN again and your program will be printed.

NAME..UT4800 DATE..05/19/80 TIME..1430 PAGE..01  
STARTING LINE..5 ENDING LINE..2055

```
5 REM ** COPYRIGHT 1979, DAKINS CORPORATION
20 GOTO 7500
24 REM ** WRITTEN BY ROD RIGGENBACH: 112779:1200
2000 TL = ( LEN (TL$) > 2): IF TL THEN FOR TL = LEN (TL$) TO 1 STEP - 1
      : IF MID$ (TL$,TL,1) = S$ THEN NEXT TL
2005 TL$ = MID$ (TL$,1,TL):TL = LEN (TL$): RETURN
2010 TL = ( LEN (PF$) < 2):T$ = "0": IF ( NOT TL) THEN FOR TL = 1 TO LEN
      (PF$) - 1: IF MID$ (PF$,TL,1) = T$ THEN NEXT TL
2015 PF$ = MID$ (PF$,TL): RETURN
2050 PH = PEEK (36) + 1:PV = PEEK (37) + 1: RETURN
2055 PF$ = MID$ (PF$,1, LEN (PF$) - PP) + "." + RIGHT$ (PF$,PP): RETURN
```

Notice that you are shown the program NAME, DATE, TIME, PAGE, plus starting and ending line numbers. Only a partial program (lines 5 through 2055) has been printed in the example.

If the program is not on the diskette inserted in the designated drive, what will be printed is a single line:

FILE NOT FOUND.

The actual printing runs under the control of an EXEC file. This makes normal error trapping impossible.

After the listing is completed, you are instructed to:

INSERT THE DISKETTE MARKED UTS003  
IN SLOT 6 DRIVE 1

Press RETURN, and you will find yourself back at the Lister screen ready to print another program. If you have no more programs to list, press the ESCape key, and you will be returned to the PROGRAMMING AIDS 3.3 MENU.

#### PRINTING AND RESET WITH THE LISTER

1. If you have not turned on your printer as asked, and you press RETURN, your printer will lock. However, it will unlock as soon as the printer is turned on, but you will have to press RETURN a second time.
2. If you decide either to limit your printing, or not to print at all for any reason, you must use the RESET key. The ESCape key is inoperative during printing with the Lister programming aid.
3. If you press RESET during the printing, and have the Autostart ROM, all you need do (after being sure that your UTS003 is inserted in drive 1) is to press RETURN. You will find yourself back at the Lister screen.
4. If you press RESET during the printing, and do not have the Autostart ROM, type 3D0G and press RETURN to put you back into Applesoft.
5. For both 3 and 4 of the above, RESET does not return control to the user. Once Applesoft is re-entered, the EXEC file, created by the Lister, continues with the rest of the commands, which are the prompt for insertion of UTS003, and the run of the Lister. The only way to avoid having the EXEC in control is to turn your Apple off and reboot the DOS.

#### ERROR MESSAGES

DISKETTE IS WRITE PROTECTED.

You are unable to update the LISTER-EXEC file. Remove the write-protect tag from your diskette.

FILE NOT FOUND.

This means that the program you requested to be listed does not exist. Verify that the data is correct before continuing. This message appears only on the printout; it is not displayed on the screen.

LISTER.EXEC IS LOCKED.

You cannot update LISTER.EXEC. Unlock LISTER.EXEC.

MENU PROGRAM NOT FOUND.

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are also prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette, or deleted the program UT0002 from the UTS003 diskette. You must restore from your backup.

**LINE CROSS  
REFERENCE**

LINE CROSS  
REFERENCE

## THE LINE CROSS REFERENCE

The Line Cross Reference either prints or displays a line number cross-reference for all lines in an Applesoft BASIC Program. You also have the option of only printing line numbers referenced in GOSUB statements.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, and 48K of memory. Use of a printer (compatible with the Apple) is optional.

You may wish to modify lines 7605 - 7615 for this program (UT4800) to be compatible with your own printer. DOS must remain in effect at all times.

To reach the Line Cross Reference program, choose Selection 2 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU. The first screen of the Line Cross Reference program shows the following data entry prompts with the dots (.) indicating the maximum response length:

TODAY'S DATE (MM/DD/YY) .....

The default is the date as entered on the PROGRAMMING AIDS 3.3 MENU. You can, however, change the date on the Cross Reference parameter screen. To correctly do so, the numerics must be entered with appropriate leading zeros, combined with slashes between month and day, and between day and year. Sample entries would be 09/10/79, and 12/03/78. The only date that remains in effect after you move from the Cross Reference to another aid, or turn off your Apple, is the date shown on the PROGRAMMING AIDS 3.3 MENU.

PROGRAM NAME

.....  
Enter the name of the program to be cross-referenced. This is actually only the name you wish to have appear at the top of your report. It is not necessary that it be the same as the name of the program that you load.

GOSUB REFERENCES ONLY? .

Replying Y to this prompt will cause the Line Cross Reference program to print only those line numbers referenced in GOSUB statements.

USING THE PRINTER OPTION? .

Are you using the printer (rather than the screen) for output? The default value is N. If you do reply Y, the following three prompts appear:

PRINTER SLOT .

This is the slot where your printer interface card is located. The default is 1.

PARALLEL INTERFACE CARD? .

If you are using a parallel interface card, accept the default value, which is Y. If you are using a serial or communications interface card, reply N.

PRINTER LINE LENGTH ...

Enter the desired line length. The permissible range is from 40 to 255 columns. The default value for printer line length is 80.

You are then asked:

IS THIS ENTRY CORRECT?

If it isn't, type N, press RETURN, and make any necessary corrections. When the entry is correct, type Y and press RETURN.

If you have selected the printer option, you are instructed to:

TURN ON PRINTER AND ALIGN PAPER.

PRESS 'RETURN' KEY TO CONTINUE.

Turn on your printer, align your paper to the top of the page, and press RETURN. The following screen will soon appear:

\*\* THE LINE CROSS REFERENCE \*\*

(C) 1979 BY DAKIN5 CORPORATION

1. LOAD PROGRAM TO BE CROSS REFERENCED.
2. CALL 35852 TO RUN CROSS REFERENCE.

YOU MAY NOW BEGIN.

]

Type LOAD and the name of the program to be line cross-referenced. Press RETURN. Then type CALL 35852 to run the Line Cross Reference. The maximum-sized program that can be cross-referenced is 124 sectors. HIMEM is set to 33792, and is not reset until you return to the Cross Reference parameter screen.

If you are not using the printer option, your program name and the date will be displayed on the screen, and you must then press RETURN to see the results of the cross-reference.

Pressing any key except RESET during the cross-reference will stop the displaying or the printing process. Pressing a key again will cause the process to continue. Pressing ESCape not only stops, but terminates the displaying or the printing. You are then instructed to insert your UTS003 diskette in Slot X Drive X to return you to the Line Cross Reference parameter screen.

If you LOAD UT6601 (The Copier), for example, and type CALL 35852, your screen at the end of the line cross-referencing will look like:

**\*\* THE LINE CROSS REFERENCE \*\***

```
7505: 4505
7520: 4524 4525 4540 4542 4545 7590
      7595 7597 7730 7820
7525: 7525 7530
7550: 7550
7560: 7560
7562: 7562
7570: 7570
7580: 7580
7586: 7586
7597: 4528 4535
7605: 4730
7607: 4526 4527 4730
7610: 4530
7730: 7660 7670 7680 7690 7700 7710
      7720 7725
7820: 7610
60000: 4700 4730 7525 7550 7560 7562
      7570 7580 7586 7595
62990: 7590 7730 62990
63000: 4730
```

PRESS 'RETURN' KEY TO CONTINUE.

What this shows is that line 7607 (as an example) is referenced by a GOSUB, GOTO, THEN, LIST or RUN on lines 4526, 4527 and 4730. Only the lines that have references are shown.

After pressing RETURN, you are told to:

INSERT THE DISKETTE MARKED UTS003  
IN SLOT 6 DRIVE 1

PRESS 'RETURN' KEY TO CONTINUE.

Press RETURN, and you will find yourself back at the Cross Reference parameter screen ready to cross-reference another program. Once at that screen, if you have no more programs to cross-reference, press the ESCape key, and you will be returned to the PROGRAMMING AIDS 3.3 MENU.

**ERROR MESSAGES**

**FILE NOT FOUND**

This means that the file you requested to be cross-referenced does not exist. Verify that the data is correct before continuing.

**I/O ERROR**

Be sure your diskette has been inserted correctly in the appropriate drive, and shows no visible damage.

## MENU PROGRAM NOT FOUND

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette or deleted the program UT0002 from the UTS003 diskette. You must restore from your back-up.

## THIS PROGRAM CANNOT BE LINE CROSS REFERENCED.

- a) This program is improperly formatted; it cannot be line cross-referenced. Programs must be properly formatted in order for the Line Cross Reference to be able to function. If they are not, the Line Cross Reference will terminate, and you will be shown the above error message. A program is improperly formatted if 1) the last line in the program does not point to the first memory location following that line, or 2) the first two memory locations following the last line of the program do not contain the null character (hex 00).

This is normally handled by Applesoft.

or

- (b) There is no Applesoft program in memory.

## TOO MANY REFERENCED LINE NUMBERS.

This program has so many GOTO, THEN, GOSUB, RUN and LIST statements that the Line Cross Reference has filled up its table of referenced line numbers (1024 references). This program is simply too large for the Line Cross Reference to handle.



VARIABLE CROSS  
REFERENCE

**MOBCE**

Look! Look! Binder Products  
424 North Cedarbrook Avenue  
St. Charles, Missouri 63025-1477 800-9332

VARIABLE CROSS  
REFERENCE

## THE VARIABLE CROSS REFERENCE

The Variable Cross Reference either prints or displays a variable name cross-reference for all variables in an Applesoft BASIC program.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, and 48K of memory. Use of a printer (compatible with the Apple) is optional.

You may wish to modify lines 7605 - 7615 for this program (UT4900) to be compatible with your own printer. DOS must remain in effect at all times.

To reach the Variable Cross Reference program, choose Selection 3 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU. The first screen of the Variable Cross Reference program shows the following data entry prompts with the dots (.) indicating the maximum response length:

### TODAY'S DATE (MM/DD/YY) .....

The default is the date as entered on the PROGRAMMING AIDS 3.3 MENU. You can, however, change the date on the Cross Reference parameter screen. To correctly do so, the numerics must be entered with appropriate leading zeros, combined with slashes between month and day, and between day and year. Sample entries would be 09/10/79, and 12/03/78. The only date that remains in effect after you move from the Cross Reference to another aid, or turn off your Apple, is the date shown on the PROGRAMMING AIDS 3.3 MENU.

### PROGRAM NAME

.....  
Enter the name of the program to be cross-referenced. This is actually only the name you wish to have appear at the top of your report. It is not necessary that it be the same as the name of the program that you load.

### USING THE PRINTER OPTION? .

Are you using the printer (rather than the screen) for output? The default value is N. If you do reply Y, the following three prompts appear:

### PRINTER SLOT .

This is the slot where your printer interface card is located. The default value is 1.

### PARALLEL INTERFACE CARD? .

If you are using a parallel interface card, accept the Y default. If you are using a serial or communications interface card, reply N.

### PRINTER LINE LENGTH ...

Enter the desired line length. The permissible range is from 40 to 255 columns. The default value for printer line length is 80.

You are then asked:

### IS THIS ENTRY CORRECT?

If it isn't, type N, press RETURN, and make any necessary corrections. When the entry is correct, type Y and press RETURN.

If you have selected the printer option, you are instructed to:

TURN ON PRINTER AND ALIGN PAPER.

PRESS 'RETURN' KEY TO CONTINUE.

Turn on your printer, align your paper to the top of the page, press RETURN, and you will soon see:

\*\* THE VARIABLE CROSS REFERENCE \*\*

(C) 1979 BY DAKIN5 CORPORATION.

1. LOAD PROGRAM TO BE CROSS REFERENCED.
2. CALL 35852 TO RUN CROSS REFERENCE.

YOU MAY NOW BEGIN.

]

Type LOAD and the name of the program to be cross-referenced. Press RETURN. Then type CALL 35852 to run the Variable Cross Reference. As long as you hear the crunching sound, the Variable Cross Reference program is finding new variables in your program.

The maximum-sized program that can be cross-referenced is 110 sectors. HIMEM is set to 30208, and is not reset until you return to the Cross Reference parameter screen.

If you are not using the printer option, your program name and the date will be displayed on the screen, and you must then press RETURN to see the results of the cross-reference.

Pressing any key except RESET during the cross-reference will stop the displaying or the printing process. Pressing a key again will cause the process to continue. Pressing ESCape not only stops, but terminates the displaying or the printing. You are then instructed to insert your UTS003 diskette in Slot X Drive X to return you to the Variable Cross Reference parameter screen.

If you LOAD UT6601 (The Copier), for example, and type CALL 35852, your screen at the end of the variable cross-referencing will look like:

\*\* THE VARIABLE CROSS REFERENCE \*\*

2355 2420 4905

WR\$  
4900

YM\$  
2420 2430 2430 2430

YY  
2435 2435 2440 2440

Z  
2000 2060 2070 2075 2075 2075 2100  
2225 2225 2225 2225 2225 2275 2300  
2300 2350 2355 2420 2430 2550 3000  
3000 3000 4905 7500

Z\$  
2075 2075 2075 2420 4905

PRESS 'RETURN' KEY TO CONTINUE.

This shows (as an example) that variable name YY occurs twice on line 2435 and twice on line 2440.

After pressing RETURN, you are instructed to:

INSERT THE DISKETTE MARKED UTS003  
IN SLOT 6 DRIVE 1  
PRESS 'RETURN' KEY TO CONTINUE.

Press RETURN, and you will find yourself back at the Cross Reference parameter screen ready to cross-reference another program. Once at that screen, if you have no more programs to cross-reference, press the ESCape key, and you will be returned to the PROGRAMMING AIDS 3.3 MENU.

### ERROR MESSAGES

#### FILE NOT FOUND

This means that the file you requested to be cross-referenced does not exist. Verify that the data is correct before continuing.

#### I/O ERROR

Be sure your diskette has been inserted correctly in the appropriate drive, and shows no visible damage.

#### MENU PROGRAM NOT FOUND.

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette or deleted the program UT0002 from the UTS003 diskette. You must restore from your backup.

#### THIS PROGRAM CANNOT BE CROSS REFERENCED.

- a) The program is improperly formatted; it cannot be cross-referenced. Programs must be properly formatted in order for the Cross Reference to be able to function. If they are not, the Cross Reference will terminate, and you will be shown the above error message. A program is improperly formatted if 1) the last line in the program doesn't point to the first memory location following that line, or 2) the first two memory locations following the last line of the program do not contain the null character (hex 00).

This is normally handled by Applesoft.

- or
- b) There is no Applesoft program in memory.

#### TOO MANY VARIABLES

This program has so many variables and references that the Cross Reference has filled up its table of referenced variable names. This program is simply too large for the Cross Reference to handle. Divide your program into two parts and run the Variable Cross Reference on each part.



PEEKER

MOBEE

Look Ltd Binder Products

424 North Cedarbrook Avenue

Springfield, Missouri 65802 14171 856-9822

PEEKER



## THE PEEKER

The Peeker displays on the screen or prints out on the printer records from random-access text files. For these files, as explained fully in the Apple DOS 3.3 manual, the length parameter must be indicated when the file is opened.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, 16K of memory if Applesoft is in ROM; 24K if not. Use of a printer is optional. You may wish to modify line 7601 to be compatible with your printer.

The Peeker is reached by choosing Selection 4 from the DAKIN5 PROGRAMMING Aids 3.3 MENU.

The screen shows one by one the following data entry prompts, with the dots indicating the maximum response length:

TODAY'S DATE (MM/DD/YY) .....

The default is the date as entered on the PROGRAMMING AIDs 3.3 MENU. You can, however, change the date on the Peeker parameter screen. To correctly do so, the numerics must be entered with appropriate leading zeros, combined with slashes between month and day, and between day and year. Sample entries would be 09/10/79 and 12/03/78. The only date that remains in effect after you move from the Peeker to another aid, or turn off your Apple, is the date shown on the PROGRAMMING AIDs 3.3 MENU.

FILE NAME

.....  
Enter the file name. There is no default value. For the printout example we have used the file name DMGL02.

RECORD LENGTH .....

The example record length is 76. This includes the carriage return at the end of the record. In order to use the Peeker, you must know the record length. The acceptable length is from 2 to 32767 characters. Whenever a record length is greater than 239, a different (and considerably slower) approach is used to read the records. Please be patient.

For your own information, it is suggested that you keep the record length as part of the file name for easy reference. You might, therefore, store DMGL02 as DMGL02.76 for convenience. The Apple DOS does not retain the record length of random-access text files anywhere on the diskette even though DOS must have that information in order to calculate where a particular record is located.

FROM: (RECORD #) ....

TO: (RECORD #) ....

FROM specifies the beginning record desired. TO specifies the ending record desired. For our example we have entered 3 as the FROM parameter and 8 as the TO parameter. Either a partial or a complete listing of the file may be obtained. The default values are 0 (FROM) and 9999 (TO) specifying that the complete file will be displayed or printed, regardless of the number of records in the file. The TO parameter cannot be less than the FROM parameter. Since record numbers for random-access text files begin with zero, 0 to 10 equals 11 records. If you created a file starting with record 1, the Peeker will show that record 0 does not exist. Any time a designated record within a specific range does not exist,

the message RECORD DOES NOT EXIST will be displayed on the screen or appear on the printout with the record number shown to the right of the message.

**SLOT .**

Enter the number of the slot in which your disk controller card is located. The default is the slot of the last drive used.

**DRIVE .**

Enter the disk drive number to be used. The default is the number of the last drive used.

**VOLUME ...**

Accept the zero (0) default if you do not use volume numbers.

**USING THE PRINTER OPTION? .**

Are you using the printer (rather than the screen) for output? The default value is N. If you do reply Y, the following three prompts appear:

**PRINTER SLOT .**

This is the slot where your printer interface card is located. The default is 1.

**PARALLEL INTERFACE CARD? .**

Parallel interface card users should accept the Y default. If you are using a serial or communications interface card, reply N.

**PRINTER LINE LENGTH ...**

The default value for printer line length is 80. The permissible range is from 40 to 255 columns.

You are asked:

IS THIS ENTRY CORRECT?

If it isn't, type N, press RETURN, and make any necessary corrections. When the entry is correct, type Y, press RETURN, and you are instructed to:

INSERT YOUR DATA DISKETTE  
IN SLOT 6 DRIVE 1

PRESS 'RETURN ' KEY TO CONTINUE.

If you have not selected the printer option, then, after you press RETURN, the file will be displayed on the next screen. The screen display will be the same as the printout except that, on the screen, the record numbers will be flashing, the first character of each record will be shown in inverse, and the column indicator line will not appear after every ten records. If the record length is greater than 256 characters, then only 256 characters at a time will be shown on the screen.

If you have selected the printer option, the following screen appears:

TURN ON PRINTER AND ALIGN PAPER.

Make sure you do turn on your printer before pressing RETURN or the printer will lock. Even if it does lock, it will unlock as soon as the printer is turned on. Align your paper to the top of a page and press RETURN. The message:

YOUR FILE IS NOW BEING LISTED.

appears, and your listing is printed.

```

FILE DMGL02  DATE 05/19/80
FROM RECORD= 3  TO RECORD= 8
1+++++10+++++20+++++30+++++40+++++50+++++60+++++70+++++8
10300, PETTY CASH, 000000000, 000000000, 000000000, 00000000000000000000 3
10400, CASH EQUIVALENTS, 000000000, 000000000, 000000000, 0000000000000000000 4
11100, ACCOUNTS RECEIVABLE-TRADE, 000000000, 000000000, 000000000, 000000000 5
11200, ACCOUNTS RECEIVABLE-EMPL, 000000000, 000000000, 000000000, 0000000000 6
11300, ALLOWANCE - BAD DEBT, 000000000, 000000000, 000000000, 00000000000000 7
11400, NOTES RECEIVABLE, 000000000, 000000000, 000000000, 000000000000000000 8

```

Notice that the number of characters displayed is one less than the specified record length since the carriage return at the end of each record is treated by DOS as part of that record. All other carriage returns within the record are displayed as commas. After the printing process is completed, you are returned to the Peeker parameter screen.

### ERROR MESSAGES

MENU PROGRAM NOT FOUND.

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette, or deleted the program UT002 from the UTS003 diskette. You must restore from your backup.

RECORD DOES NOT EXIST.

The designated record within the range specified does not exist.

UNABLE TO FIND FILE XXXXXX.

The file to be displayed or printed could not be found on the data diskette. Verify the file name, and that the correct diskette is in the correct drive. Press RETURN to be taken to the Peeker parameter screen.

UNABLE TO OPEN FILE XXXXXX.

The file to be displayed or printed could not be opened. Either the file is not a text file or an I/O error occurred while trying to open the file. Press RETURN to be taken to the Peeker parameter screen.



PATCHER

MCBEE

Loose Leaf Binder Products

424 North Cedarbrook Avenue

Springfield, Missouri 65802 (417) 866-0822

PATCHER

## THE PATCHER

The Patcher permits you to display and update data in any sector of a diskette. This program can prove to be an interesting and instructive aid in the hands of a hobbyist who is a relative newcomer to personal computers, as well as being an extremely effective tool for the most skilled and sophisticated of users.

For the purpose of updating, you may use either the approach of entering the name of a program or file listed in the catalog, or of identifying a specific track/sector combination.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, 48K of memory, and DOS 3.3.

The Patcher is reached by choosing Selection 5 from the PROGRAMMING AIDS 3.3 MENU. The first screen of the Patcher program shows the following data entry prompts:

FILE NAME

.....  
with the dots (.) indicating the maximum response length. This is the first of two different approaches to updating that are possible with the Patcher programming aid.

With this first approach, you enter the name of any file or program listed in the catalog. For example, enter UT6601. This is the Copier program. Press RETURN, and the next data entry prompt that appears is:

SLOT .

This is the slot in which your disk controller card is located. The default value is the slot of the last drive used.

DRIVE .

The program UT6601 resides on your Programming Aids 3.3 diskette. If your utilities diskette is in drive 1, press RETURN and accept the default which is 1.

VOLUME ...

Accept the 0 default if you do not use volume numbers.

Your screen should look like:

\*\* THE PATCHER \*\*

FILE NAME

UT6601

SLOT 6

DRIVE 1

VOLUME 0

IS THIS ENTRY CORRECT?

After entering Y to indicate that the entry is correct, press the RETURN key, and you will be instructed to:

INSERT YOUR DATA DISKETTE  
IN SLOT 6 DRIVE 1

PRESS 'RETURN' KEY TO CONTINUE.

WARNING: Place a write-protect sticker on your Programming Aids 3.3 diskette while learning to use the Patcher.

After inserting your Programming Aids 3.3 diskette as required, press the RETURN key. In front of you will appear the following screen:

\*\* THE PATCHER \*\*

```
TRACK 09    SECTOR 0E
00- 67 1B 2C 08 05 00 B2 2A . . . . 2*
08- 2A 20 43 4F 50 59 52 49 * COPYRI
10- 47 48 54 20 31 39 37 39 GHT 1979
18- 2C 20 44 41 4B 49 4E 35 , DAKINS
20- 20 43 4F 52 50 4F 52 41  CORPORA
28- 54 49 4F 4E 00 36 08 14 TION 6. .
30- 00 AB 37 35 30 30 00 70 . +7500. .
38- 08 D0 07 54 4C D0 28 E3 . P. TLP(
40- 28 54 4C 24 29 CF 5A 29 (TL$)02)
48- 3A AD 54 4C C4 81 54 4C :-TLD. TL
50- D0 E3 28 54 4C 24 29 C1 P. (TL$)A
58- 31 C7 C9 31 3A AD EA 28 1G11:-. (
60- 54 4C 24 2C 54 4C 2C 31 TL$, TL, 1
68- 29 D0 53 24 C4 82 54 4C )PS$D. TL
70- 00 90 08 D5 07 54 4C 24 . . . U. TL$
78- D0 EA 28 54 4C 24 2C 31 P. (TL$, 1

UPDATE :
```

What is displayed is the first half of sector 0E with the cursor positioned at the bottom of the screen to the right of the "UPDATE:" prompt. Notice on the right the lines of BASIC that are represented in hexadecimal form on the left. This shows you how BASIC programs are stored in memory.

The Patcher will display one by one every sector of data contained in that file, offering you the opportunity to make as many changes as you wish. At this point, just keep pressing RETURN to view the different sectors. When you have examined numerous sectors, press the ESCape key, and you will see the screen:



\*\* THE PATCHER \*\*

NO UPDATE MADE.

PRESS 'RETURN' KEY TO CONTINUE.

This means that you have not made any changes to any of the sectors in the program UT6601. Press RETURN, and you will find yourself back at the initial data entry screen.

The second approach enables you to identify a specific track/sector combination. Two examples of use would be to change the name of the "HELLO" program name on your diskette (track 1, sector 9), and to find and correct control characters in program names on directory sectors. To use this option, simply press the RETURN key in response to the FILE NAME prompt. Then, specify the slot, and either accept the default of 1 for drive, or enter 2. Reply to the volume prompt and press RETURN. When asked if the entry is correct, reply Y. The next screen will instruct you to:

INSERT YOUR DATA DISKETTE  
IN SLOT 6 DRIVE 1

PRESS 'RETURN' KEY TO CONTINUE.

After inserting your data diskette in the appropriate drive, press RETURN, and you will be asked:

PLEASE ENTER PATCH TRACK #

PLEASE ENTER PATCH SECTOR #

The range for a track must be between 0 and 34 (00-22 in hexadecimal form), and for a sector between 0 and 15 (00-0F hex). All track and sector numbers must always be entered in hexadecimal form. There are 35 tracks on each diskette, and 16 sectors within each track. Each sector contains 256 bytes of data.

Following the second approach, enter track 11, and sector 0. Your screen will appear as:

PLEASE ENTER PATCH TRACK #11

PLEASE ENTER PATCH SECTOR #0

Press RETURN and you will be shown the following:

\*\* THE PATCHER \*\*

```
TRACK 11    SECTOR 00
00- 04 11 0F 03 00 00 FE 00 .....
08- 00 00 00 00 00 00 00 00 .....
10- 00 00 00 00 00 00 00 00 .....
18- 00 00 00 00 00 00 00 00 .....
20- 00 00 00 00 00 00 00 7A .....
28- 00 00 00 00 00 00 00 00 .....
30- 20 01 00 00 23 10 00 01 ...#.
38- 00 00 00 00 00 00 00 00 .....
40- 00 00 00 00 00 00 00 00 .....
48- 00 00 00 00 00 00 00 00 .....
50- 00 00 00 00 00 00 00 00 .....
58- 00 00 00 00 00 00 00 00 .....
60- 00 00 00 00 00 00 00 00 .....
68- 00 00 00 00 00 00 00 00 .....
70- 00 00 00 00 00 00 00 00 .....
78- 00 00 00 00 00 00 00 00 .....

UPDATE :
```

This is the section of DOS that contains the VTOC or Volume Table of Contents. For more information concerning the material before you on the screen, reference the section covering the VTOC in your Apple DOS 3.3 (Disk Operating System Instruction and Reference Manual).

The left side of the display shows the relative hex addresses within the sector of the first byte shown on each line beginning at \$00, and ending at \$78. The middle portion of the screen displays the hex values for each byte within the sector. At the top left-hand side of the screen you see address 00. You will note that this byte has a hex value of 04. As you proceed across the screen to your right, observe that address 01 is equal to 11, 02 equal to 0F, and 06 equal to FE.

The right portion of the screen displays the character equivalents for the hex values. Dots "." are displayed if no printable character equivalents exist. Press the RETURN key to view the second half of the sector. Press RETURN again to return to the track/sector prompt.

Another interesting track/sector combination to examine is track 11, sector 0F. This is the Diskette Directory, also discussed in the same DOS 3.3 manual. It will appear on your screen as:

\*\* THE PATCHER \*\*

TRACK 11      SECTOR 0F

```
00- 00 11 0E 00 00 00 00 00 .....
08- 00 00 00 14 0F 82 D5 D4 ..... UT
10- B0 B0 B0 B2 A0 A0 A0 A0 0002
18- A0 A0 A0 A0 A0 A0 A0 A0
20- A0 A0 A0 A0 A0 A0 A0 A0
28- A0 A0 A0 A0 16 00 16 0F .....
30- 82 D5 D4 B1 B5 B0 B0 A0 .UT1500
38- A0 A0 A0 A0 A0 A0 A0 A0
40- A0 A0 A0 A0 A0 A0 A0 A0
48- A0 A0 A0 A0 A0 A0 A0 1D
50- 00 18 0F 82 D5 D4 B1 B6 ..... UT16
58- B0 B0 A0 A0 A0 A0 A0 A0 00
60- A0 A0 A0 A0 A0 A0 A0 A0
68- A0 A0 A0 A0 A0 A0 A0 A0
70- A0 A0 21 00 1B 0F 82 D5 !.... U
78- D4 B2 B6 B0 B0 A0 A0 A0 T2600
```

UPDATE :

Regardless of which approach you use, the Patcher will display the first half of a sector, and then prompt you for any modifications you wish to make. After you have finished modifying this half of a sector, simply press the RETURN key in response to the update prompt at the bottom of the screen, and you will be shown the second half of the sector.

Using either of the aforementioned two approaches, three types of modifications are possible: 1) Hexidecimal Modify, 2) Character Modify, and 3) Display Modify.

Hexidecimal Modify allows you to enter two-digit hexadecimal numbers to replace each byte of data in a sector.

Character Modify permits you to enter alpha, numeric, and special character data from the keyboard, and have each character you enter replace a byte in the sector. In the Character Modify mode, the high order bit of each byte is set to zero. Therefore, the letters A-Z will be placed in the sector as hex values \$41-\$5A. BASIC programs are stored in this format.

Display Modify will let you enter alpha, numeric, and special character data from the keyboard. Each character entered will replace a byte in the sector. In this mode, the high order bit of each byte is set to a one. Therefore, the letters A-Z will be placed into the sector as hex values \$C1-\$DA. Text files are stored in this format.

### Hexidecimal Modify - Format/Example

Following the second approach, that of track/sector combination, display track 11, sector 0F on your screen. This is the Diskette Directory sector. The format for entering data after "UPDATE:" (at the bottom of the screen) is:

```
adrs:Xdata data data
```

"adrs" is a two-digit hexadecimal number giving the relative hex address (from the beginning of the sector) of the first byte to be modified. This address can be determined from the addresses at the left of the Patcher display screen. For example, the addresses of the top line (track 11, sector 0F) are:

```
00 01 02 03 04 05 06 07
```

and for the second line:

```
08 09 0A 0B 0C 0D 0E 0F
```

The "X" tells the Patcher that you wish to do a Hexidecimal Modification.

"data" is a two-digit hexadecimal number giving the value to be put into the sector. Each "data" portion is separated from the others by a space. For example, if you wish to change the data at address \$76 from 82 to 02, you would Enter

```
76:X02
```

and your screen should look like:

```
** THE PATCHER **
```

```
TRACK 11    SECTOR 0F

00- 00 11 0E 00 00 00 00 00 .....
08- 00 00 00 14 0F 82 D5 D4 .....UT
10- B0 B0 B0 B2 A0 A0 A0 A0 0002
18- A0 A0 A0 A0 A0 A0 A0 A0
20- A0 A0 A0 A0 16 00 16 0F
28- A0 A0 A0 A0 16 00 16 0F
30- 82 D5 D4 B1 B5 B0 B0 A0 .UT1500
38- A0 A0 A0 A0 A0 A0 A0 A0
40- A0 A0 A0 A0 A0 A0 A0 A0
48- A0 A0 A0 A0 A0 A0 1D
50- 00 18 0F 82 D5 D4 B1 B6 ....UT16
58- B0 B0 A0 A0 A0 A0 A0 A0 00
60- A0 A0 A0 A0 A0 A0 A0 A0
68- A0 A0 A0 A0 A0 A0 A0 A0
70- A0 A0 21 00 1E 0F 82 D5 !....U
78- D4 B2 B6 B0 B0 A0 A0 A0 T2600

UPDATE :76:X02
```

You might wish to change not just \$76 but \$77 and \$78 as well. Your change could be:

```
76:X02 D1 D2
```

leaving a space between data entries after 02. At this time, only make the 76:X02 change on the screen. Press RETURN once. Now you will see your update reflected on the screen as:

\*\* THE PATCHER \*\*

```
TRACK 11    SECTOR 0F
00- 00 11 0E 00 00 00 00 00 .....
08- 00 00 00 14 0F 82 D5 D4 ..... UT
10- B0 B0 B0 B2 A0 A0 A0 A0 0002
18- A0 A0 A0 A0 A0 A0 A0 A0
20- A0 A0 A0 A0 A0 A0 A0 A0
28- A0 A0 A0 A0 16 00 16 0F .....
30- 82 D5 D4 B1 B5 B0 B0 A0 .UT1500
38- A0 A0 A0 A0 A0 A0 A0 A0
40- A0 A0 A0 A0 A0 A0 A0 A0
48- A0 A0 A0 A0 A0 A0 A0 1D
50- 00 18 0F 82 D5 D4 B1 B6 .... UT16
58- B0 B0 A0 A0 A0 A0 A0 A0 00
60- A0 A0 A0 A0 A0 A0 A0 A0
68- A0 A0 A0 A0 A0 A0 A0 A0
70- A0 A0 21 00 16 0F 02 D5 !.... U
78- D4 B2 B6 B0 B0 A0 A0 A0 T2600

UPDATE :
```

If you were to press RETURN again, you would be shown the second half of the sector. Pressing RETURN once more would cause the sector to be written out to diskette. To avoid having this change go into effect, press the ESCape key, and you will be told: NO UPDATE MADE.

The effect of writing the 76:X02 change to diskette would be to unlock UT2600.

Character Modify - Format/Example

The format for entering data after "UPDATE:" is:

adrs:Ccharchar

"adrs" is the same as for the Hexidecimal Modify. "C" tells the Patcher that you wish to do a Character Modify. "char" is a single character typed on the keyboard.

No spaces are allowed between characters. If you type in a space, it is treated as any other character. To see a demonstration of Character Modify, follow the first approach, and for FILE NAME type in UT4800. Then enter the slot number, the number of the drive containing your Programming Aids 3.3 diskette, and volume number, if applicable. You will see the first half of track 21, sector E.

\*\* THE PATCHER \*\*

TRACK 21      SECTOR 0E

```
00- 64 15 2C 08 05 00 B2 2A ..... 2*
08- 2A 20 43 4F 50 59 52 49 * COPYRI
10- 47 48 54 20 31 39 37 39 GHT 1979
18- 2C 20 44 41 4B 49 4E 35 , DAKINS
20- 20 43 4F 52 50 4F 52 41 CORPORA
28- 54 49 4F 4E 00 36 08 14 TION. 6. .
30- 00 AB 37 35 30 30 00 65 .+7500. .
38- 08 18 00 B2 2A 2A 20 57 ... 2** W
40- 52 49 54 54 45 4E 20 42 RITTEN B
48- 59 20 52 4F 44 20 52 49 Y ROD RI
50- 47 47 45 4E 42 41 43 48 GGENBACH
58- 3A 20 31 31 32 37 37 39 : 112779
60- 3A 31 32 30 30 00 9F 08 :1200...
68- D0 07 54 4C D0 28 E3 28 P. TLPC. (
70- 54 4C 24 29 CF 5A 29 3A TL$)OZ):
78- AD 54 4C C4 81 54 4C D0 -TLD. TLP
```

UPDATE :

For a Character Modify, change the date and time: 112779:1200, located on lines 12 and 13. The date and time data extends from \$5A-\$64. Enter

5A:C010280.0915

Press the RETURN key, and your change will be displayed as:

\*\* THE PATCHER \*\*

TRACK 21      SECTOR 0E

```
00- 64 15 2C 08 05 00 B2 2A ..... 2*
08- 2A 20 43 4F 50 59 52 49 * COPYRI
10- 47 48 54 20 31 39 37 39 GHT 1979
18- 2C 20 44 41 4B 49 4E 35 , DAKINS
20- 20 43 4F 52 50 4F 52 41 CORPORA
28- 54 49 4F 4E 00 36 08 14 TION. 6. .
30- 00 AB 37 35 30 30 00 65 .+7500. .
38- 08 18 00 B2 2A 2A 20 57 ... 2** W
40- 52 49 54 54 45 4E 20 42 RITTEN B
48- 59 20 52 4F 44 20 52 49 Y ROD RI
50- 47 47 45 4E 42 41 43 48 GGENBACH
58- 3A 20 30 31 30 32 38 30 : 010280
60- 2E 30 39 31 35 00 9F 08 .0915...
68- D0 07 54 4C D0 28 E3 28 P. TLPC. (
70- 54 4C 24 29 CF 5A 29 3A TL$)OZ):
78- AD 54 4C C4 81 54 4C D0 -TLD. TLP
```

UPDATE :

Press the ESCape key to prevent this change from being written out to diskette.

Display Modify - Format/Example

The format for entering data after "UPDATE:" is:

adrs:Dcharcharchar

"adrs" is the same as for the Hexidecimal Modify. "D" tells the Patcher that you wish to do a Display Modify. "char" is a single character typed on the keyboard. As with the Character Modify, no superfluous spaces are permitted. If you type in a space, it is treated the same as any other character. For a demonstration of Display Modify, following the first approach, for FILE NAME enter LISTER.EXEC. Enter the appropriate slot, drive and volume information. Then, after indicating that the entry is correct, insert your Programming Aids 3.3 diskette into the designated drive, and you will be shown the first sector of this file.

NOTE: The track and sector numbers (as well as the contents) of the LISTER.EXEC will change each time the Lister program is run.

\*\* THE PATCHER \*\*

TRACK 20      SECTOR 03

```
00- C9 CE D4 8D C8 C9 CD C5 INT.HIME
03- CD BA AD B2 B7 B3 B9 B2 M:-27392
10- 8D B1 C3 C1 CC CC AD B7 .ICALL-7
13- B4 B1 BA D0 CF CB C5 B3 41:POKE3
20- B4 AC B2 B2 BA C3 C1 CC 4.22:CAL
28- CC AD B9 B3 B6 BA C5 CE L-936:EN
30- C4 8D D2 D5 CE 8D CE C5 D.RUN.NE
38- D7 8D CC CF C1 C4 D3 C1 W.LOADS
40- CD D0 CC C5 AC D3 B6 AC MPLE.56,
48- C4 B1 AC D6 B0 8D CC C9 D1.V0.LI
50- D3 D4 B0 AC B6 B3 B9 B9 ST0.6399
58- B9 8D D0 D2 A3 B0 8D B0 9.PR#0.0
60- D6 D4 C1 C2 B1 B0 BA D0 VTB10:P
68- D2 C9 CE D4 A2 87 A0 A0 RINT".
70- A0 A0 C9 CE D3 C5 D2 D4    INSERT
78- A0 D4 C8 C5 A0 C4 C9 D3    THE DIS
```

UPDATE :

You are going to change the data located at the addresses \$3A-\$3D from LOAD to RUN. Since RUN is one space shorter than LOAD, you will add a space after the N to have a total of four characters. After "UPDATE:" enter

3A:DRUN

and press the RETURN key. Your change will be displayed as:

\*\* THE PATCHER \*\*

TRACK 20      SECTOR 03

```
00- C9 CE D4 8D C8 C9 CD C5 INT. HIME
08- CD BA AD B2 B7 B3 B9 B2 M:-27392
10- 8D B1 C3 C1 CC CC AD B7 .1CALL-7
18- B4 B1 BA D0 CF CB C5 B3 41:POKE3
20- B4 AC B2 B2 BA C3 C1 CC 4,22:CAL
28- CC AD B9 B3 B6 BA C5 CE L-936:EN
30- C4 8D D2 D5 CE 8D CE C5 D. RUN. NE
38- D7 8D D2 D5 CE A0 D3 C1 W. RUN SA
40- CD D0 CC C5 AC D3 B6 AC MPLE, S6,
48- C4 B1 AC D6 B0 8D CC C9 D1, V0. LI
50- D3 D4 B0 AC B6 B3 B9 B9 5T0, 6399
58- B9 8D D0 D2 A3 B0 8D B0 9. PR#0. 0
60- D6 D4 C1 C2 B1 B0 BA D0 VTAB10:P
68- D2 C9 CE D4 A2 87 A0 A0 RINT".
70- A0 A0 C9 CE D3 C5 D2 D4 INSERT
78- A0 D4 C8 C5 A0 C4 C9 D3 THE DIS
```

UPDATE :

Press the RETURN key again, and you will be shown the second half of the sector. To have the sector written out to diskette (which you do not want to do), press the RETURN key again. Otherwise, press the ESCape key, and the RUN change will not go into effect.

Each UPDATE prompt is always ended by pressing the RETURN key. This same half of the sector will be re-displayed immediately reflecting your update. You may enter any number of characters or two-digit hexadecimal numbers that you wish, and each successive "char" or "data" element will replace the next byte in the sector beginning at "adrs". Any "char" or "data" entered that would cause the Patcher to attempt to update a byte past "adrs" \$FF will be ignored. Anytime you do not follow the correct format, you will receive the error message INVALID FORMAT, and you will be prompted again with

UPDATE:

You may enter as many updates for a given sector as you wish. By pressing the RETURN key only, in response to the "UPDATE:" prompt, you will be shown the second half of the sector (if bytes \$00 through \$7F were on the screen). If you press RETURN again (and you had entered a file name), the first half of the next sector of the file will be shown. If you did not enter a file name, then you will be prompted to enter another track/sector combination.

If you had entered a valid update request, the updated sector would be written out to diskette before the next sector is displayed. To avoid having a given sector updated, you may simply press the RETURN key after each half of the sector is displayed, or if you have entered a valid update request for this sector,



you may press the ESCape key in response to the "UPDATE:" prompt. This will return you to the main program with a NO UPDATE MADE message. At that point, follow the instructions on the screen, and PRESS 'RETURN' TO CONTINUE. You will then be shown the initial Patcher screen.

## QUICK REFERENCE GUIDE FOR WRITING OUT TO DISKETTE WHEN USING THE PATCHER

---

Regardless of which approach you use, or which type of modify you intend to make, you are always presented with the first half of a sector. Before any update is written out to diskette, you are forced to view both halves of a sector. Study the following situations:

1. You are shown the first half of sector X. You enter an update and press RETURN. The same half of sector X will appear reflecting your change. The change has not yet been written out to diskette. Press RETURN, and the second half of the sector will appear. Press RETURN again, and your update for the first half of the sector is written out to diskette.
2. You are shown the first half of Sector X. You press RETURN, and the second half of that sector is displayed. You make an update and press RETURN. The same second half of sector X will appear reflecting your change. The change has not yet gone into effect. It is still subject to your approval. Press ESCape, and you will receive the message NO UPDATE MADE. If you press RETURN (instead of ESCape) the update will be written out to diskette.

### ERROR MESSAGES

#### INVALID FORMAT

You must enter data in accordance with the format outlined in the documentation. Re-enter data following the correct format.

#### MENU PROGRAM NOT FOUND

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette, or deleted the program UT002 from the UTS003 diskette. You must restore from your back-up.

#### RANGE ERROR

The range for a track must be between 0 and 34 (00-22 hex); the range for a sector must be between 0 and 15 (00-0F hex). Re-enter in accordance with the designated range limitations.

#### THE DISKETTE IS WRITE PROTECTED.

You have a write-protect sticker on your diskette. Remove the sticker and start over.

#### THIS SECTOR IS EMPTY.

When using the track/sector approach to updating, you requested that the Patcher display a sector that cannot be read. Verify that your diskette has been inserted correctly, and shows no apparent physical damage.

UNABLE TO FIND FILE XXXXXX.

The above named file could not be found on the data diskette. Check to see that the correct diskette is in the correct drive. Control is returned to the initial Patcher screen.

UNABLE TO READ AFTER THE UPDATE.

After a sector is written out to the diskette, it is read and compared with the original input data. This error message means that the Patcher program was unable to do the read. Be sure your diskette has been inserted correctly in the appropriate drive, and shows no visible physical damage before attempting to update the sector again.

UNABLE TO READ THE DISKETTE.

The Patcher was unable to read the diskette. Verify that your diskette has been inserted correctly, and shows no apparent physical damage.

UNABLE TO VERIFY THE UPDATE.

After a sector is written out to the diskette, it is read and compared to the original input data. The Patcher program found a discrepancy when it compared the original data to what had been written out to the diskette.

UNABLE TO WRITE THE DISKETTE.

The Patcher was unable to write out to the diskette. Be sure your diskette has been inserted correctly, and shows no visible physical damage.

**COPIER**

COPIER

## THE COPIER

The Copier will copy any file or program from one diskette to another. The only information needed is the name of the file or program.

### Hardware Requirements:

1 Apple II computer (booted by DOS 3.3), 2 disk drives, and 48K of memory.

The Copier is reached by choosing Selection 6 from the PROGRAMMING AIDS 3.3 MENU.

The Copier shows the following data entry prompts, with the dots (.) indicating the maximum response length:

### FILE NAME

.....  
Enter the file or program name. There is no default value.

### SOURCE SLOT .

This is the slot where your disk controller card is located. The default value is the slot of the drive last used.

### SOURCE DRIVE .

This is the source drive into which you will insert the diskette you will copy from. The default value is the drive last used.

### SOURCE VOLUME ...

Accept the  $\emptyset$  default value if you do not use volume numbers.

### DESTINATION SLOT .

The default slot for the destination diskette is the same as for the source diskette. You may change this if you wish, but we do not recommend it. This places an overload on your power supply and may damage your computer.

### DESTINATION DRIVE .

This is the destination drive into which you will insert the diskette you will copy to. If you have entered 1 for the source drive, the default for the destination drive will automatically be 2.

### DESTINATION VOLUME ...

Accept the zero ( $\emptyset$ ) default if you do not use volume numbers.

You are next asked:

IS THIS ENTRY CORRECT?

You may answer Y or N only. If there is an item that you want to change, answer N, and the cursor returns to the first item which is FILE NAME.

It is very simple to make changes since you need only to press RETURN to accept the default value for any item. In this case, the defaults represent the data (accepted as valid) you last entered.

When you are satisfied that all the information is correct, reply Y to the IS THIS ENTRY CORRECT? prompt, and press RETURN. The following screen will appear:

**\*\* THE COPIER \*\***

INSERT YOUR SOURCE DISKETTE  
IN SLOT 6 DRIVE 1.

INSERT YOUR DESTINATION DISKETTE  
IN SLOT 6 DRIVE 2.

PRESS 'RETURN' KEY TO CONTINUE.

After you insert your diskettes and press RETURN, you are told:

YOUR FILE IS NOW BEING COPIED.

followed by the flashing message:

DO NOT INTERRUPT!

If you accidentally press the RESET key, turn off your Apple, or pull the plug from the wall socket, you must consider your copy as unsuccessful (although your source data remains unaltered). Start your copying process again.

When your file or program has been copied successfully, the message:

COPY COMPLETED.

will appear. Press the RETURN or ESCape key, and once again, you will be shown the FILE NAME prompt. This will permit you to copy a second file or program to another diskette. If you have no more files to copy, press ESCape to return to the PROGRAMMING AIDS 3.3 MENU.

**ERROR MESSAGES**

DESTINATION DISKETTE HAS 105 FILES.

The maximum number of files that can exist on one diskette is 105. An attempt has been made to add file number 106.

DESTINATION DISKETTE IS FULL.

There is no more space available on the destination diskette. If this happens, it is as though the copying process never started for this file. You must either use another destination diskette or delete files to make more space before you restart the copy.

DISKETTE IN DRIVE 2 IS WRITE PROTECTED.

You have a write-protect sticker on the destination diskette. Remove and restart the copy.

MENU PROGRAM NOT FOUND.

The DAKIN5 PROGRAMMING AIDS MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette or deleted program UT0002 from the UTS003 diskette. You must restore from your backup.

#### SINGLE DRIVE COPY IS NOT POSSIBLE.

The Copier program requires two drives to copy data from one diskette to another. If you specified a different volume number for the destination volume than you did for the source volume, the Copier allows the drives to be the same on the assumption that you have a hard disk, and wish to copy from one volume to another. Attempting this with a diskette will result in a VOLUME MISMATCH message.

#### THIS FILE IS LOCKED ON DRIVE 2.

The file being copied already exists on drive 2, and is locked. You must unlock this file before you restart the copy. The Copier will delete the file on drive 2 before beginning the actual copying process.

#### UNABLE TO FIND FILE XXXXXX.

The file being copied could not be found on the source diskette.

#### UNABLE TO READ DESTINATION DRIVE 2.

After a sector is written out to the destination diskette, it is read and compared with the original input data. This error message means that the Copier program was unable to do the read. Be sure your diskettes have been inserted correctly in the appropriate drives, and show no visible physical damage before attempting to perform the copy again.

#### UNABLE TO READ SOURCE DRIVE 1.

The Copier program was unable to read the source diskette. Verify that it has been inserted correctly, and shows no apparent physical damage.

#### UNABLE TO VERIFY DESTINATION DRIVE 2.

After a sector is written out to the destination diskette, it is read and compared to the original input data. The Copier program found a discrepancy when it compared the source diskette data to what had been written to the destination diskette.

#### UNABLE TO WRITE DESTINATION DRIVE 2.

The Copier program was unable to write to the destination diskette in drive 2. Verify that the diskette has been inserted correctly, and shows no apparent damage.

#### VOLUME MISMATCH.

The volume number on either the source diskette or the destination diskette did not agree with the volume number you entered.





DISKETTE COPY

MCBEE

Loose Leaf Binder Products

424 North Cedarbrook Avenue

Springfield, Missouri 65902 (417) 866-0822

DISKETTE COPY

## THE DISKETTE COPY

The Diskette Copy is an intelligent data transfer program. It initializes the output diskette, copies each track, and verifies that the output is exactly the same as the input. The Diskette Copy program should be used first to copy your Dakin5 Programming Aids 3.3 diskette. This is a precautionary measure to ensure that you have a backup system.

### Hardware Requirements:

1 Apple II computer, 2 disk drives, 48K of memory, and DOS 3.3.

To run the Diskette Copy program, choose Selection 7 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU.

The Diskette Copy screen shows you one by one the following data entry prompts, with the dots indicating the maximum response length:

### SOURCE SLOT: .

This is the slot in which your disk controller card is located. The default is the slot of the last drive used.

### DRIVE: .

This is the source drive in which you will insert the diskette you will copy from. The default is the number of the last drive used.

### VOLUME: ...

Accept the  $\emptyset$  default if you do not use volume numbers.

### DESTINATION SLOT: .

The default slot for the destination diskette is the same as for the source diskette. You may change this if you wish, but we do not recommend it. This places an overload on your power supply and may damage your computer.

### DRIVE: .

This is the destination drive into which you will insert the diskette you will copy to. If you have entered 1 for the source drive, the default for the destination drive will automatically be 2.

### VOLUME: ...

This is the volume number that you wish to have on your new copy. The default value of  $\emptyset$  tells the Diskette Copy program to use whatever volume number exists on the source diskette as the volume number for the destination diskette. The exception to this rule is when you specify you wish only to initialize. In that instance, the  $\emptyset$  default creates a volume number of 254 on the destination diskette.

### DO YOU WANT TO INITIALIZE ONLY? .

Answering Y to this question will create a properly initialized diskette without DOS, but will also not copy any data from the source diskette. Accept the default N, and you are prompted:

### DO YOU WANT TO COPY DOS? .

If you do not accept the Y default, then tracks zero, one, and two will not be copied regardless of the information they contain. This will provide you with 32 sectors of additional storage. (Track  $\emptyset$  is not available for use.)

The next question is:

IS THIS ENTRY CORRECT?

You may answer Y or N only. If there is an item that you want to change, answer N, and the cursor returns to the first item which is SOURCE SLOT.

It is very simple to make changes since you need only to press RETURN to accept the default value for any item. In this case, the defaults represent the data (accepted as valid) you last entered.

When you are satisfied that all the information is correct, reply Y to the IS THIS ENTRY CORRECT? prompt, and press RETURN. The following screen will appear:

\*\* DISKETTE COPY \*\*

INSERT YOUR SOURCE DISKETTE  
IN SLOT 6 DRIVE 1

INSERT YOUR DESTINATION DISKETTE  
IN SLOT 6 DRIVE 2

PRESS 'RETURN' KEY TO CONTINUE.

Insert both of your diskettes at this time even if you are only going to initialize the destination diskette. Toward the end of the initialization, the source diskette is read for a brief moment in order to get a pattern for the Volume Table of Contents (VTOC) on the destination diskette.

As soon as you press RETURN the following screen appears:

YOUR DISKETTE IS NOW BEING COPIED.  
DO NOT INTERRUPT!

The source diskette is first verified (unless you chose to INITIALIZE ONLY) to ensure that the diskette, and its data to be copied, are not damaged. This is an important procedure, especially when you are copying to your only backup diskette, since the next step of initialization destroys all data on the destination diskette. After initializing the output diskette, the copying process begins with the red light appearing on first one and then the other drive, verifying the input data against the output data until all the information has been copied. The initialization process does not add DOS to the destination diskette. However, if DOS exists on the source diskette, it will be copied to the destination diskette unless you specified that you did not want the Diskette Copy program to copy DOS.

It may seem to you that the copy takes a long time. To copy a diskette will take three minutes. This is due to verification, initialization, and the extra time required to verify the output data.

During the copying process the ESCape key is inoperative. Do not interrupt the copying at any time. If the power goes off, you turn off your computer, the plug is pulled from the socket, or the RESET key is pressed, you should consider your destination diskette data as invalid. Restart the copy process.

At the end of a successful, uninterrupted copy the following screen will appear:

COPYING COMPLETED.

PRESS 'RETURN' KEY TO CONTINUE.

To copy more diskettes, press RETURN, and you are again presented with the screen instructing you to insert your source and destination diskettes in the appropriate drives in preparation for another copy. If you wish to do another copy, but want to change such parameters as slot, drive, etc., press ESCape instead of RETURN. When you have no more diskettes to copy, press the ESCape key twice to return to the PROGRAMMING AIDS 3.3 MENU.

#### ERROR MESSAGES

##### MENU PROGRAM NOT FOUND.

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette or deleted the program UT0002 from the UTS003 diskette. You must restore from your backup.

##### SINGLE DRIVE COPY IS NOT POSSIBLE.

The Diskette Copy program requires two drives to copy data from one diskette to another. If you specified a different volume number for the destination volume than you did for the source volume, the Diskette Copy allows the drives to be the same on the assumption that you have a hard disk, and wish to copy from one volume to another. Attempting this with a diskette will result in an UNABLE TO VERIFY SOURCE DRIVE message.

##### UNABLE TO INITIALIZE DESTINATION DRIVE.

The copy program does an initialization of the destination diskette before beginning the actual copy. This error means that the initialization was unsuccessful or the destination diskette is write-protected. Try another diskette. This error message will also appear after approximately 25 seconds if no diskette has been inserted in the destination drive.

NOTE! Even if initialization is successful, it does not put a copy of DOS on the destination diskette.

##### UNABLE TO READ DESTINATION DRIVE.

After a track is written out to the destination diskette, it is read and compared with the original input data. This error message means that the copy program was unable to do the read. Be sure your diskettes have been inserted correctly in the appropriate drives, and show no visible physical damage before attempting to perform the copy again.

##### UNABLE TO WRITE DESTINATION DRIVE.

The copy program was unable to write to the destination diskette.

##### UNABLE TO VERIFY DESTINATION DRIVE.

This means that the copy program found a discrepancy when it compared the original source diskette data to what had been written to the destination diskette.

##### UNABLE TO READ SOURCE DRIVE.

The copy program was unable to read the source diskette, or had a volume mismatch.

##### UNABLE TO VERIFY SOURCE DRIVE.

The Copy program was unable to read the source diskette during the verify.



ARRAY EDITOR

MCBE

Loose Leaf Binder Products

424 North Cedarbrook Avenue

Springfield, Missouri 65802 (417) 856-0222

ARRAY EDITOR





\*\* ADD TO ARRAY \*\*

- 1. ....
- .....
- .....

For explanatory purposes, enter TEST ONE and press RETURN. This will take you to the line for item 2. Type TEST TWO and press RETURN. Follow this same pattern through entry seven, except to type TEST SEVEN for the line that is numbered 6, and TEST SIX for the line that is numbered 7. This will give you something to correct later on. After entering TEST SIX, and before pressing the RETURN key, your screen will appear as:

\*\* ADD TO ARRAY \*\*

- 3. TEST THREE
- 4. TEST FOUR
- 5. TEST FIVE
- 6. TEST SEVEN
- 7. TEST SIX.....
- .....
- .....

These entries were short, using only part of the response field, but whenever you reach the maximum response length for an item, as indicated by the dots after TEST SIX, you must press RETURN for the next consecutive number to appear. The maximum number of items that can be entered is 200, at which time you will see a new ADD TO ARRAY screen.

\*\* ADD TO ARRAY \*\*

200 ENTRIES HAVE BEEN USED.

SELECT THE 'END' OPTION AND SAVE IN  
ORDER TO PURGE ANY REMOVED ENTRIES.

PRESS 'RETURN' KEY TO CONTINUE.

You must then save this file and recall it in order to purge any entries that you may have removed. This will provide more space for adding more entries. If no entries have been removed, then you must save this file and begin a new one.



PRINTER LINE LENGTH . . .

This only appears if you replied Y to the parallel interface card prompt. The default value for printer line length is 80. The permissible range is from 40 to 255 columns.

You are next asked:

IS THIS ENTRY CORRECT?

You may answer Y or N only. If there is an item that you want to change, answer N, and the cursor returns to the first item which is TODAY'S DATE.

It is a very simple and quick task to make changes since you need only to press RETURN to accept the default value for any item. In this case, the defaults represent the data (accepted as valid) you last entered.

When you are satisfied that all the information is correct, reply Y to the IS THIS ENTRY CORRECT? prompt, and the next screen will instruct you to turn on the printer and to align the paper. Press RETURN, and you are informed that your report is now being prepared. Your printed result will appear as:

TESTING NUMBERS

DATE 05/19/80  
PAGE 1

TEST ONE  
TEST TWO  
TEST THREE  
TEST FOUR  
TEST FIVE  
TEST SEVEN  
TEST SIX

If you choose Selection 2 from the LIST ARRAY MENU, your entries will be numbered as follows:

TESTING NUMBERS

DATE 05/19/80  
PAGE 1

1. TEST ONE  
2. TEST TWO  
3. TEST THREE  
4. TEST FOUR  
5. TEST FIVE  
6. TEST SEVEN  
7. TEST SIX

You may interrupt the printing process or screen display at any time by pressing the ESCape key. You will be returned to the LIST ARRAY MENU which will show an additional selection, CONTINUE PROCESSING.



**\*\* CHANGE ARRAY \*\***

```
WHICH ENTRY (LAST IS 7) 7
7.TEST SIX.....
.....
.....
```

Change the word SIX to SEVEN. After making the change, press RETURN. Then press ESCape to return to the ARRAY EDITOR MENU.

If you press the ESCape key prior to pressing the RETURN key, you will also find yourself at the ARRAY EDITOR MENU. The difference being that your change will not be recorded. At times, this can prove a useful procedure whenever you decide not to make a change after all.

When you are back at the ARRAY EDITOR MENU, you find you need to delete entry 2. Choose Selection 3, REMOVE FROM ARRAY, as found on the ARRAY EDITOR MENU. Once at the REMOVE FROM ARRAY screen, enter 2, press RETURN, and your screen will appear as:

**\*\* REMOVE FROM ARRAY \*\***

```
WHICH ENTRY (LAST IS 7) 2
2. TEST TWO
```

REMOVE THIS ENTRY? .

Reply Y, and press the RETURN key for the entry to be removed from the file. The cursor then returns to the WHICH ENTRY prompt to allow you to remove additional entries. If you had typed in a negative reply to the REMOVE THIS ENTRY? prompt, your entry would not have been deleted. You would, however, still have been returned to the WHICH ENTRY question, enabling you to change the number of the entry to be deleted. Pressing ESCape at any point prior to replying Y to the REMOVE THIS ENTRY? query returns you to the ARRAY EDITOR MENU.

On the assumption that you have replied Y to the REMOVE THIS ENTRY? question, press RETURN again. The 2 default will be accepted and the following screen (containing the error message ENTRY DOES NOT EXIST.) will appear:

**\*\* REMOVE FROM ARRAY \*\***

```
WHICH ENTRY (LAST IS 7) 2
```

```
ENTRY DOES NOT EXIST.
PRESS 'RETURN' KEY TO CONTINUE.
```

This is a simple way of verifying that your entry really was removed.  
NOTE: The remaining entries in the file will be renumbered starting at one when the file is saved on diskette. Consequently, after the SAVE, your former entry 3 will become entry 2.

Now, press ESCape to return to the ARRAY EDITOR MENU.

Choose Selection 5, END, to save your newly created text file. You will see the following screen:

**\*\* SAVE ARRAY \*\***

SAVE AS .....

You enter here the name by which you want the array to be saved. For this example, type TESTING NUMBERS and press RETURN. You should be aware of the fact that whenever the same name is used for saving two different text files, the last is recorded, and the first deleted. (If you did not want to save this array, you would press the space bar, and then the RETURN key, at which point you would be asked to RECALL another array.) You are now instructed to supply the following data entry information:

SLOT .

This is the slot where your disk controller card is located. The default value is the slot of the last drive used.

DRIVE .

Enter the number of the disk drive to be used. In this case, type 2 to save the array on drive 2. The default value is the number of the drive last used.

VOLUME ...

Accept the 0 default value if you do not use volume numbers.

You are then asked:

IS THIS ENTRY CORRECT?

If it isn't, type N, press RETURN, and make any necessary corrections. When the entry is correct, type Y, press RETURN, and you are instructed to:

INSERT YOUR DATA DISKETTE  
IN SLOT 6 DRIVE 2

PRESS 'RETURN' KEY TO CONTINUE.

Into the designated drive, insert the diskette (not UTS003) on which you wish to have the array saved, and press the RETURN key. The following message will appear on the screen:

YOUR APPLE IS NOW COMPUTING.  
DO NOT INTERRUPT!

and the array is written out to diskette. You are then given the opportunity to recall an array.

You can either recall an array, press RETURN in order to be taken to the ARRAY EDITOR MENU, or press ESCape, which will return you to the DAKIN5 PROGRAMMING AIDS 3.3 MENU.

Since there was no available space on the DAKIN5 PROGRAMMING AIDS 3.3 diskette, the file TESTING NUMBERS had to be saved on another diskette in drive 2. You now find yourself at the first Array Editor parameter screen which shows the following data entry prompts, with the dots (.) indicating the maximum response length:

RECALL ARRAY .....

Enter the title TESTING NUMBERS of your newly created array.

SLOT .

Enter the slot number in which your controller card resides. The default value is the slot of the last drive used.

DRIVE .

Enter the disk drive to be used. TESTING NUMBERS was saved on drive 2. The default is the number of last drive used.

VOLUME ...

Accept the 0 default value if you do not use volume numbers.

You are then asked:

IS THIS ENTRY CORRECT?

If it isn't, type N, press RETURN, and make any necessary corrections. Once you are satisfied the entry is correct, type Y, press RETURN, and you are instructed to:

INSERT YOUR DATA DISKETTE  
IN SLOT 6 DRIVE 2

PRESS 'RETURN' KEY TO CONTINUE.

Insert the diskette containing the array to be recalled, and press the RETURN key. The following message will appear on the screen:

YOUR FILE IS NOW BEING LOADED.

The array will be read in, and then the ARRAY EDITOR MENU screen will appear. When you have finished making any additions or changes, save your array, and ESCape back to the PROGRAMMING AIDS 3.3 MENU.

NOTE: For an example of an EXEC file created with the Array Editor, once you are at the RECALL ARRAY screen, type in DATA ENTRY. Indicate the slot and drive where your Programming Aids 3.3 diskette is located. Then when you are presented with the LIST ARRAY MENU, either display or print this EXEC file, and notice that colons and commas are acceptable characters. At the SAVE ARRAY screen, press ESCape to be returned to the PROGRAMMING AIDS 3.3 MENU, since the EXEC file you have been examining is a locked file and permanently saved on the UTS003 diskette.



## ERROR MESSAGES

### ARRAY FILE IS INVALID.

The Array file you have tried to recall is not a properly formatted text file for use with the Array Editor. Press RETURN, and re-enter a valid text file name, or press RETURN twice to begin work on a new file. Only sequential-access text files with fewer than 91 characters between carriage returns can be used with the Array Editor.

### DATE IS INVALID.

You must enter a valid calendar date for the LIST ARRAY screen. Re-enter the date correctly.

### DISKETTE IS FULL.

There is no more space available on the diskette in the designated drive. Use another diskette, or delete files to make more space available.

### ENTRY DOES NOT EXIST.

For any of the ADD, CHANGE, or REMOVE FROM ARRAY selections, you must not exceed the parameter of the data entry prompt as indicated, or attempt to remove or change an entry that has already been removed. Re-enter a number within the indicated range.

### FILE IS LOCKED ON DRIVE X.

The array you have attempted to save already exists on your diskette, and is locked.

### MENU PROGRAM NOT FOUND.

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette or deleted the program UT0002 from the UTS003 diskette. You must restore from your back-up.

### NO ENTRIES TO CHANGE.

No entries have been entered. Choose Selection 1, ADD TO ARRAY, from the ARRAY EDITOR MENU.

### NO ENTRIES TO LIST.

No entries have been entered. Choose Selection 1, ADD TO ARRAY, from the ARRAY EDITOR MENU.

### NO ENTRIES TO REMOVE.

No entries have been entered. Choose Selection 1, ADD TO ARRAY, from the ARRAY EDITOR MENU.

200 ENTRIES HAVE BEEN USED. SELECT THE 'END' OPTION AND SAVE IN ORDER TO PURGE ANY REMOVED ENTRIES.

The maximum allowable size for the file has been reached. This file size limit includes all entries that have been removed since the file was last saved to diskette. To get rid of these removed entries and compress the file, it is necessary to save the array to diskette, and then recall the array.

UNABLE TO FIND FILE XXXXXX.

The file to be recalled could not be found on your data diskette. Verify the name of the file, and that the correct diskette is in the correct drive. Press RETURN to show again the RECALL ARRAY screen.

CALCULATOR

MCBE

Loose Leaf Binder Products

424 North Cedarbrook Avenue

Springfield, Missouri 65802 (417) 866-0822

CALCULATOR

## THE CALCULATOR

The Calculator program performs addition, subtraction, multiplication and division. The Calculator subroutine (using twenty place accuracy) is written in Assembler code, and handles numeric string data many times faster than an equivalent Applesoft BASIC subroutine.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, and 48K of memory.

To run the Calculator demonstration program, choose Selection 9 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU. The Calculator screen shows one by one the following data entry prompts, with the dots indicating the maximum response length:

ADDEND .....

Enter up to twenty digits to be added. When the number is negative, the minus sign takes the place of the twenty-first position (if the maximum number of digits has been entered). The default is 0. The minus sign always follows the digits entered.

ADDEND .....

Type in up to twenty more digits to be added. As stated with the first addend, the default is 0. If a minus sign is included, it must immediately follow the digits already appearing on the screen, and it occupies the twenty-first position (if the maximum number of digits has been entered).

Your screen should now look something like the following:

```
** THE CALCULATOR **  
ADDEND 12345678998765432109  
ADDEND 9876543210987654321-
```

Press RETURN, and you see displayed:

```
** THE CALCULATOR **  
ADDEND 12345678998765432109  
ADDEND 9876543210987654321-  
246913578777777788  
CALCULATOR ROUTINE ENDED.  
PRESS 'RETURN' KEY TO CONTINUE.
```

The number printed in the center of the screen is the sum of the addition.

The second demonstration of the Calculator is subtraction. The default values for both the minuend and subtrahend are 0, and as many as twenty digits can be entered for each, with the twenty-first position reserved for the minus sign (if the maximum number of digits has been entered). After you complete the subtraction, your screen should appear as:

\*\* THE CALCULATOR \*\*

MINUEND 99999999998765432123

SUBTRAHEND 21987654321234567123

78012345677530865000

CALCULATOR ROUTINE ENDED.

PRESS 'RETURN' KEY TO CONTINUE.

The third demonstration of the Calculator is multiplication. The default values for the multiplicand and multiplier are 0. The conditions and parameters are the same as those for the addition and subtraction examples. At the end of the demonstration, your screen might show a product forty digits long, plus a minus sign, as in the following illustration:

\*\* THE CALCULATOR \*\*

MULTIPLICAND 12345678998765432123

MULTIPLIER 98765432198765432198-

1219326322100289592262155160350327696354

CALCULATOR ROUTINE ENDED.

PRESS 'RETURN' KEY TO CONTINUE.

For the division demonstration of the Calculator, the conditions and parameters are the same as those for addition, subtraction, and multiplication, with the exception that there is no default value for the divisor. Your division exercise completed, you should see:

**\*\* THE CALCULATOR \*\***

DIVIDEND 98765432198765432198

DIVISOR 3456

28577960705661294

134

CALCULATOR ROUTINE ENDED.

PRESS 'RETURN' KEY TO CONTINUE.

The first number printed below the divisor is the quotient, followed by the remainder.

Pressing RETURN at this point will show you the first screen for the ADDITION demonstration. To return to the PROGRAMMING AIDS 3.3 MENU, press ESCape at any time during the Calculator demonstration program.

Detailed information concerning the use of the Calculator's routine follows:

CALCULATOR ROUTINE

This subroutine will add, subtract, multiply or divide two signed or unsigned numeric strings, and place the result in a third string. The result of the calculations will be right justified with leading zeros removed. A minus sign will be added to the right of a negative numeric result.

The user is responsible for defining all three strings before calling the Calculator routine. The result string must be of at least two characters long, and must be (for addition/subtraction) at least as big as each of the other two strings. For multiplication and division, the result must be big enough to hold the result of the calculation. For example, if you wish to add A\$ to B\$ giving C\$ and A\$ = "923" and B\$ = "17", then C\$ must be at least three characters long. However, if B\$ = "275", then C\$ must be at least four characters long to handle the overflow. If you wish to multiply A\$ by B\$ giving C\$, and A\$ = "923" and B\$ = "17", then C\$ must be at least 5 characters long. However, if B\$ = "275", then C\$ must be at least six characters long to handle the overflow.

To Call the Calculator Routine

1. BLOAD CALC.OBJ
2. Set HIMEM: 36864
3. Establish values for the two strings to be added, subtracted, multiplied or divided, and assign space for the result string.

4. Call the subroutine with the following format:

```
For Add - CALL 36864:A$B$C$
For Subtract - CALL 36877:A$B$C$
For Multiply - CALL 36890:A$B$C$
For Divide - CALL 36903:A$B$C$
```

B\$ will be added to, subtracted from, multiplied by, or divided into A\$, and the result will be placed in C\$. The variable names after the call can be any one- or two-character numeric strings in your program. The colon must be there. The C\$ may be followed by another colon, or may be at the end of the line. You will not set a syntax error when executing this statement because the Calculator routine causes BASIC to skip over the A\$B\$C\$. After using division, A\$ will contain the remainder. You must make sure that B\$ does not equal zero if you are using division.

Example:

```
10 PRINT CHR$(4);"BLOAD CALC.OBJ":HIMEM:36864
20 A$="456"
30 BX$="123"
40 CA$="00000":REM COULD BE CA$="  "
50 CALL 36890:A$BX$CA$
60 PRINT CA$
```

Running this program would cause the number 56088 to be printed.

If line 50 were CALL 36903:A\$BX\$CA\$ then the number 3 would be printed, and A\$ would contain the remainder of 87.

The CALC.OBJ program begins at address \$9000 and has a length of \$600.

NOTE: Any syntax errors encountered by the Calculator routine will cause an appropriate error code to be set, after which the Calculator will return to your BASIC program. You can then check the error code by a PEEK(38399). If the Calculator found no errors, this location would contain a zero.

## ERROR MESSAGES

### ERROR CODES

#### 1 IMPROPER SYNTAX IN CALL.

Either the call was not followed by a colon, or the colon was not followed by three string variable names.

#### 2 INPUT VARIABLE IS WRONG LENGTH.

Either the length of one of the three string variables was greater than 20 (the result string may be 40 characters long for multiplication), or the length of the 3rd variable was less than the length of one of the other two variables.

#### 3 INPUT VARIABLE IS NOT DEFINED.

One of the three string variables was not defined in the program prior to the call.

#### 4 INPUT VARIABLE IS NOT NUMERIC.

One of the first two variables did not contain numeric data.



SCREEN PRINTER

MCBEE

Loose Leaf Binder Products

424 North Cedarbrook Avenue

Springfield, Missouri 65802 (417) 886-0822

SCREEN PRINTER

## THE SCREEN PRINTER

The Screen Printer allows you to print any normal text appearing on the screen, provided the cursor is visible, and, therefore, the keyboard active. Inverse and flashing characters will be printed as normal characters.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, 16K of memory if Applesoft is in ROM; 24K if not. 1 Printer (compatible with the Apple). You may wish to modify line 7599 to be compatible with your printer. DOS must remain in effect at all times.

The Screen Printer is reached by choosing Selection 10 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU.

The Screen Printer screen shows the following data entry prompts with the dots (.) indicating the maximum response length:

### PRINTER SLOT .

This is the slot where your printer interface card is located. The default value is 1.

### DOES PRINTER GENERATE LINE FEED? .

The cursor appears over Y, which is the default value. If your printer or interface card does generate a line feed when it encounters a carriage return, press the RETURN key to accept the default. If neither your printer nor your interface card generates a line feed, reply N, press RETURN, and an automatic line feed character will be sent to your printer after each carriage return.

### COMMUNICATIONS INTERFACE CARD? .

If you are not using a communications interface card for your printer, press the RETURN key to accept the N default.

### IS THIS ENTRY CORRECT? .

If the entry is not correct, type N, press RETURN, and make any necessary corrections. When the entry is correct, type Y, and press the RETURN key.

The next screen will instruct:

TURN ON THE PRINTER.

PRESS 'RETURN' KEY TO CONTINUE.

NOTE: This printer message will not appear if you answered that you have a communications interface card.

Press RETURN as requested, and the next screen will prompt:

PRESSING THE CTRL-Z KEY WILL NOW SEND  
WHAT IS ON THE SCREEN TO THE PRINTER.

DON'T FORGET TO TURN ON THE PRINTER.

PRESS 'RETURN' KEY TO CONTINUE.

The screen printer program is now operative. Press the RETURN key, and you will be returned to the PROGRAMMING AIDS 3.3 MENU.

Anytime the cursor is visible, and you want to print the contents of the screen, hold down the CTRL key and the Z key simultaneously. Printing will start immediately without pressing RETURN. You may print the same screen as many times as you like by simply repeating the CTRL-Z combination since you remain at the same screen during, and after printing. Please note that one position (line 1, column 1) of the screen will not be printed, but will be recorded as blank on the printout.

Make sure that your printer is turned on before you exercise the CTRL-Z combination, or the system will "hang up" on you. You may then either have to press the RESET key, or turn off your computer and reboot the system.

The Screen Printer program remains in effect until you press RESET or reboot the system regardless of what is on the screen or which program is running. However, the Screen Printer code is located at memory locations \$30A-\$3CF and might get overlaid by some assembler subroutine in your program. DOS remains in effect after the Screen Printer program has been activated. If your program changes the contents of the DOS Input register or the Apple Monitor Input register, then the Screen Printer will no longer be operative. Refer to the section in Apple's DOS Version 3.3 manual which covers the subject of selecting I/O devices.

#### ERROR MESSAGES

MENU PROGRAM NOT FOUND.

The DAKIN5 PROGRAMMING AIDS 3.3 MENU could not be found on the diskette UTS003. You are prompted to insert UTS003 and try again. This would be an unusual situation where you may have damaged the UTS003 diskette or deleted the program UT0002 from the UTS003 diskette. You must restore from your backup.

PROMPTER

**MOBE**

Loose Leaf Binder Products

424 North Clearbrook Avenue

Springfield, Missouri 65802 (417) 886-0922

PROMPTER

## THE PROMPTER

The Prompter is a demonstration program that illustrates the power of a data entry subroutine used in Dakin5 Corporation programs. This data entry subroutine should also prove extremely useful to you for your own programs.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, and 16K of memory if Applesoft is in ROM; 24K if not.

The Prompter is reached by choosing Selection 11 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU.

The Prompter allows you to enter on the screen values for the various parameters that are required by the data entry subroutine. Then the Prompter sets up these parameters, and calls the data entry subroutine just as you would in your own program.

The screen shows one by one the following data entry prompts, with the dots indicating the maximum response length:

### PROMPT STRING .....

This is the message that you would print in your program before calling the data entry subroutine to accept a user response to this prompt. The cursor is also assumed to be flashing in the first position of the response field prior to calling the data entry subroutine. For our example, we have used CUSTOMER BALANCE.

### AT COLUMN (1-13) ..

This is the HTAB value that you would set in your own program before printing the prompt string. For our example, we have used 3, indicating that the string "CUSTOMER BALANCE" will be printed starting in horizontal position 3 of the screen. Your numbers in Column (1-13) will vary depending on the length of the PROMPT STRING.

### AT LINE (3-21) ..

This is the VTAB value that you would set in your program before printing the prompt string. For our example, we have used 10, indicating that the string "CUSTOMER BALANCE" will be printed on line 10 of the screen.

### RESPONSE LENGTH (0-11) ..

This is the maximum response length that the data entry subroutine will allow. You set this parameter yourself for your program before calling the data entry subroutine. You do this by setting the variable PL equal to the maximum response length you desire. For our example, we have used 10, indicating that the user response to our CUSTOMER BALANCE prompt can be no more than 10 characters long. You will notice 10 positions are available on sample screen two after CUSTOMER BALANCE, e.g., 12345.....

### RESPONSE TYPE (0=ALPHANUMERIC 1=NUMERIC + OR - 2=NUMERIC + ONLY 3=USER DEFINED) .

These are the response types that the data entry subroutine permits. RESPONSE TYPE 0 means that all alphabetic, numeric, and special characters will be accepted as valid with the exception of:

1. COMMA ",,"
2. COLON ":"
3. CONTROL CHARACTERS: all characters combined with the Control key. Exceptions are CTRL M, X, H, U, and the ESCape key which have special valid accepted functions.

- CTRL M - terminates the input in the same way as a carriage return.
- CTRL X - resets the response field to its original state, and places the cursor in the first position of the response field.
- CTRL H - (or the left arrow) erases the last character typed, and replaces it with the character that was there before.
- CTRL U - (or the right arrow) copies the character to the right of the cursor as if the character had been typed.
- ESCape - terminates the input in exactly the same way as the carriage return. In addition, it sets an error flag (E=1) which your program can test after returning from the data entry subroutine. NOTE: The ESCape key can be treated as an invalid response character if the data entry subroutine input parameter "RE" is set to 1 rather than 0.

These conditions of commas, colons, and control characters also apply to RESPONSE TYPES 1 and 2.

RESPONSE TYPE 1 means that only numeric characters will be accepted as valid. The minus sign will also be accepted as valid only after at least one numeric character has been entered. After entering the minus sign, no additional characters can be entered.

RESPONSE TYPE 2 means that only positive (+) numeric characters will be accepted as valid. As mentioned, all conditions of commas, colons, and control characters (outlined in categories 1-3 of exceptions to accepted valid data) pertain to RESPONSE TYPES 0, 1, and 2. In addition, for all three types, a bell sounds if an attempt is made to enter an invalid character, and the cursor does not advance.

RESPONSE TYPE 3 means that you will define the characters that are to be accepted as valid.

You should set the response type parameter in your program before calling the data entry subroutine. This is done by setting the variable PT equal to 0, 1, 2, or 3. For our example, we have used 2 to indicate that a numeric unsigned response is desired.

With a RESPONSE TYPE=3 situation, you would then be prompted for VALID CHARACTER SET before seeing the ORIGINAL DEFAULT information request. You would enter the characters you wish to have accepted as valid by the data entry subroutine.



If you chose RESPONSE TYPES 1 or 2, the following data entry prompts will also appear on the Prompter's initial parameter screen:

#### DISPLAY FORMAT:

##### DECIMAL PLACES .

This is the number of decimal places the data entry subroutine should assume in this response. If a number greater than 0 is used for this parameter, the data entry subroutine will insert a period into the response field in the proper position. You should set this parameter in your own program before calling the data entry subroutine. You do this by setting the variable PP equal to a value in the range 0 to 99 inclusive. A value of zero means that no decimal point is to be inserted. For our example, we have used 2, indicating that the last two digits entered in response to the CUSTOMER BALANCE prompt will appear to the right of the decimal point.

##### INCLUDE COMMAS? .

This indicates whether or not the data entry subroutine should insert commas into the response field. If commas are to be inserted, they appear after every three digits starting from the decimal point (or rightmost digit if no decimal point exists), and proceeding to the left. You should set this parameter in your program before calling the data entry subroutine. You do this by setting the variable PC equal to 0 if you do not want commas, and equal to 1 if you do want commas. For our example, we have indicated that we do want commas by replying Y.

If you chose RESPONSE TYPE 2, an additional data entry prompt will be shown after INCLUDING COMMAS?.

##### LEADING ZEROS? .

This indicates whether or not the data entry subroutine should add leading zeros to the response in order to make the response length equal the maximum response length established in variable PL. You should establish this parameter in your program before calling the data entry subroutine. You do this by setting the variable PZ equal to 1 if you want leading zeros, and equal to 0 if you don't want zeros. In our example, we have indicated by entering Y that we do want zeros.

At this point, regardless of the response type selected, the following data entry prompt will appear on the Prompter screen:

#### ORIGINAL DEFAULT

.....

The number of dots for the ORIGINAL DEFAULT will equal the RESPONSE LENGTH parameter. ORIGINAL DEFAULT is the value that the data entry subroutine will display next to the prompt string. Using a default enhances your prompt, allowing a value to be entered by simply pressing the RETURN key. This is also a parameter which should be set before calling the data entry subroutine. This is done by setting the variable RD\$ equal to the default value to be used for this prompt, or by setting the RD\$ equal to the null string (RD\$="") if no default value is to be displayed. In our example, we have chosen a default value of 12345. Using our entry data, your screen would appear as follows:

**\*\* THE PROMPTER \*\***

PROMPT STRING CUSTOMER BALANCE

AT COLUMN (1-13) 3

AT LINE (3-21) 10

RESPONSE LENGTH (0-11) 10

RESPONSE TYPE (0=ALPHANUMERIC

1=NUMERIC + OR -

2=NUMERIC + ONLY

3=USER DEFINED) 2

DISPLAY FORMAT:

DECIMAL PLACES 2

INCLUDE COMMAS? Y

LEADING ZEROS? Y

ORIGINAL DEFAULT

12345

You are then asked:

IS THIS ENTRY CORRECT?

If it isn't, type N, and the cursor will return to the first item of the parameter screen PROMPT STRING. Make necessary corrections until you are satisfied with the information entered. Then reply Y to the prompt IS THIS ENTRY CORRECT?

The next screen will appear

CUSTOMER BALANCE 12345.....

showing the prompt string (and the original value if you entered one) located on the column and line that you requested. The data entry subroutine displays the default value and awaits your response. Only valid characters according to the response type selected can now be entered.

After entering the data you wish, press RETURN. The data entry routine will now "echo" the response by displaying it in the response field with all necessary decimal points, commas, leading zeros, and right justification as requested. RESPONSE TYPES 1 and 2 are always right justified. RESPONSE TYPES 0 and 3 are always left justified.

You will then be prompted to:

PRESS 'RETURN' KEY TO CONTINUE.

When you press RETURN, the cursor will go back to the first position of the response field. At that time you may enter more data. To return to the Prompter screen, press the ESCape key. At the parameter screen you can enter other data entry subroutine parameters, or press the ESCape key again to return to the PROGRAMMING AIDS 3.3 MENU.

In our example, we accepted the default value by pressing RETURN. What was originally:

CUSTOMER BALANCE                    12345.....

is displayed as:

CUSTOMER BALANCE                    00,000,123.45

PRESS 'RETURN' KEY TO CONTINUE.

Notice that, although the maximum response length was 10, thirteen characters are now displayed in the response field. The extra three characters came from the two commas and the period (decimal point). You need to take this into account when using the various options of the data entry subroutine, and make sure to allow enough room between the prompt string and the response field.

#### DATA ENTRY SUBROUTINE

Input parameters\* to be set before calling the data entry subroutine:

<u>Variable Name</u>	<u>Meaning</u>
PL	Maximum Response Length. Must be in the range of 0 to 100 inclusive. When PL is 0, only the carriage return and the ESCape key may be typed. This can be used for a "PRESS 'RETURN' KEY TO CONTINUE." type prompt.
PT	Response Type =0 Alphanumeric input. =1 Numeric input with possible minus sign. =2 Numeric input with no sign. =3 User defined (see RT\$).
PP	Number of Decimal Places =0 No decimal places Maximum value is 99.
PC	Comma Insert Flag =0 No commas are to be inserted. =1 Insert commas every third digit (before the decimal point).
PZ	Zero-Fill Flag =0 Remove leading zeros. =1 Add leading zeros to fill up the response to its maximum length (PL).
RD\$	Default String. Set to the default value to be displayed in the response field, or set to null string if no default is to be displayed.

\*The values of all these input variables remain unchanged after calling the data entry subroutine.

Variable NameMeaning

RE

Escape Protect Flag  
 =0 ESCape key may be used to terminate a response.  
 =1 ESCape character is treated as an invalid response character.

RT\$

Set of valid characters. It must be a string of all characters to be accepted as valid by the data entry subroutine. If PT is set to 3, then the data entry subroutine will compare each character typed with all of the characters in string RT\$.

Work variables used by the data entry subroutine:

TL\$	TL	T\$
TC\$	T	TX\$
TY\$	TM	TH

Output parameters returned from the data entry routine are:

Variable NameMeaning

RS\$

The Response String. If a carriage return was the last character typed, then RS\$ equals the response up to, but not including, the carriage return.  
 If the carriage return was the only character typed, then RS\$ equals the default string (RD\$) that was input to the data entry routine.  
 If the ESCape was the last character typed, and the input variable RE equals 0, then RS\$ equals the null string.  
 NOTE: RS\$ does not include any of the commas or the decimal point used by the data entry routine to echo the response (see PF\$).  
 However, RS\$ will contain leading zeros, and LEN (RS\$) will equal the maximum response length (PL) if the input variable PZ was set to one.

E

Escape Pressed Flag  
 =0 The ESCape key was not the last key pressed.  
 =1 The ESCape key was the last key pressed.

R

Last Character Value. It contains the ASCII value of the last character typed.  
 =13 RETURN key was typed.  
 =27 ESCape key was typed.

Variable Name	Meaning
RS\$	Last Character typed. It would be either the RETURN character or the ESCape character.
PF\$	The Formatted String. This is the formatted response that is echoed by the data entry subroutine including decimal point and commas.
PH	First column (HTAB) of response field.
PV	Line (VTAB) on which response was entered.
PX	Number of extra characters needed to re-display (echo) the response. This is a count of the decimal point, and the number of commas required to display a response of maximum length (PL).

The following code demonstrates how your program might use the data entry subroutine to accomplish the same thing that we did in the example using the Prompter demonstration program. The data entry subroutine is located at line number 2250. After the GOSUB 2250 has been completed, RS\$ will contain the response value.

```
7500 GOSUB 4900
7510 TEXT : HOME : VTAB 10: HTAB 3: PRINT "CUSTOMER BALANCE      ";
7520 PL = 10: PT = 2: PP = 2: PC = 1: PZ = 1: RD$ = "12345": RE = 0: GOSUB 2250
63999 END : REM DATA ENTRY SUBROUTINE
```

Try this yourself. Return to the DAKIN5 PROGRAMMING AIDS 3.3 MENU, and press ESCape. At the top of the screen appears the following banner:

```
      DAKIN5 PROGRAMMING AIDS 3.3 COMPLETED.
      ]
```

with the Applesoft symbol displayed. EXEC DATA ENTRY from the Dakin5 Programming Aids 3.3 diskette. It is this EXEC file that will create the data entry subroutine. Enter lines 7510 and 7520 as shown. Then run the program. You are free to begin writing your own program at line 7510.

The GOSUB 4900 initializes some general purpose parameters as well as some that are required by the data entry subroutine. This GOSUB must be executed once before attempting to call the data entry subroutine.

There is one additional feature of the data entry subroutine that you can use. By changing the GOSUB 2250 to a GOSUB 2150, the response will not be redisplayed on the screen with commas, leading zeros, etc. RS\$ will still contain the response upon return from the GOSUB. You may wish to use the GOSUB 2150 for selection prompts and Yes/No prompts.

NOTE: If response type 3 is being used, make sure that the most frequently entered characters are at the beginning of string RT\$. This will make the validation faster. Also, to have a double quote as one of the valid characters, define RT\$ as all of the other valid characters, such as RT\$= "1234" and then RT\$=RT\$+CHR\$(34).



**CRUNCHER**

**MORE**

**Loose Leaf Binder Products**

**424 North Crumback Avenue**

**Springfield, Missouri 65802-1477 866-0822**

CRUNCHER



## THE CRUNCHER

The Cruncher compresses code in Applesoft programs. It also removes unreferenced (dead) code, unreferenced REM (remark) statements and the comment after the REM in referenced REM statements. The resulting code will run faster and consume less space (on disk and in memory) than the original program.

### Hardware Requirements:

1 Apple II computer, 1 disk drive, and 48K of memory.

By using the Cruncher, you may save more than 45% of your memory and disk space.

With the Cruncher, you must LOAD and SAVE your own programs to be compressed.

The only way to exit from the Cruncher program is to CALL 36402 as indicated on the Cruncher screen; the ESCape key is inoperative.

To reach the Cruncher program, choose Selection 12 from the DAKIN5 PROGRAMMING AIDS 3.3 MENU. You will see the following screen:

\*\* THE CRUNCHER \*\*

(C) 1979 by DAKIN5 CORPORATION

1. LOAD THE PROGRAM TO BE COMPRESSED
2. CALL 36361 TO RUN THE CRUNCHER
3. SAVE THE COMPRESSED PROGRAM
4. CALL 36402 TO RETURN TO PROGRAMMING AIDS 3.3 MENU

Simply type LOAD and the name of the program to be crunched. Programs on UTS003 have been completely crunched; so LOAD one of your own programs. Then CALL 36361 to start the Cruncher. Listen to the geiger counter-type sound as each line of the program is compressed. The longer you hear the sound, the more memory you are saving.

After the program has been compressed, you can save it if you wish. For example, you might type SAVECOPYCR to indicate the crunched version of the COPY program, if that were the program you had loaded. To save the crunched program under the same name as the original, you should first delete the original before doing the save. This is because DOS does not make the additional space available when a long program is overlaid by a shorter one.

WARNING: The Cruncher will frequently compress code so much that certain lines will be too long for you to modify. We suggest that you keep the original version of your programs as well as the compressed version.

If you have numerous programs to be compressed, follow steps 1-3 as many times as needed. You never lose the instructions for the Cruncher due to internal setting of the screen window. Only the commands you enter will gradually disappear off the screen.

When you wish to return to the PROGRAMMING AIDS 3.3 MENU, type CALL 36402.

The maximum size program that the Cruncher is able to compress is 127 sectors. HIMEM is set to 34303 by the Cruncher, and is not reset until you return to the PROGRAMMING AIDS 3.3 MENU.

Literals used in PRINT or DATA statements must be enclosed in quotation marks. For example, use

```
10 PRINT "ABC" instead of
10 PRINT ABC
```

NOTE: REM statements beginning with an asterisk will be left in the program.

To determine the exact memory savings produced by the Cruncher, follow these steps:

1. After you LOAD your program and press RETURN, you will see the Applesoft symbol ] . Enter: PRINT PEEK (175)+PEEK(176)\*256 and press RETURN. Record the number that appears on the screen.

2. After you CALL 36361 and press RETURN, enter again the following: PRINT PEEK (175)+PEEK(176)\*256 and press RETURN. Record the number that appears on the screen. The difference between the two recorded numbers is the number of bytes of memory saved by the Cruncher.

#### ERROR MESSAGES

CANNOT COMPRESS.

a) Programs must be properly formatted in order for the Cruncher to be able to compress them. If they are not, the program will terminate, and you will be shown the above error message. A program is improperly formatted if 1) the last line in the program doesn't point to the first memory location following that line, or 2) the first two memory locations following the last line of the program do not contain the null character (hex 00). This is normally handled by Applesoft.

or

b) There is no Applesoft program in memory.

TOO MANY LINES.

The program being compressed has so many GOTO, THEN, GOSUB, LIST or RUN statements that the Cruncher has filled up its table of referenced line numbers (1024 references). This program is simply too large for the Cruncher to handle.



