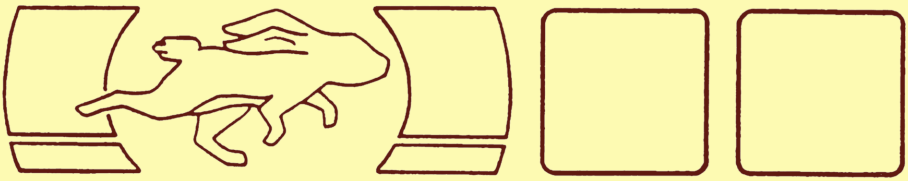
WorkBook71

SPREADSHEET
FILE MANAGER
REPORT GENERATOR

for the **HP-71**

by Richard E. Harvey



WorkBook71

SPREADSHEET
FILE MANAGER
REPORT GENERATOR

for the **HP-71**

by Richard E. Harvey

SERIAL NUMBER 2702-ET-020

CONTENTS

Introduction

WorkBook71 is a data-gathering, analysis- and report-generating tool for the Hewlett-Packard HP-71 Portable Computer. Modules of the system include Editor, data format converter, searching, and Report Formatter. Each module can operate separately of others in the group and, by virtue of data format conversion, other programs you may have can be used interactively. The goal has been to provide a versatile package in which data can flow between applications with minimal hassle.

Data is organized in a grid of columns and rows. The intersection of a column and row is called a cell. Data stored in these cells forms the building blocks of a data analysis system known as an Electronic Spreadsheet.

Cells can contain information such as names and addresses, numbers and formulas. This flexibility makes it easy for us to maintain a mailing list, travel expense report, budget or business statement and write reports in essentially the same format. Once you are comfortable with the spreadsheet concept you will probably find it easier to write a file to solve a problem than writing a BASIC program.

Some conventions are accepted for files written on a portable computer. We want to keep memory usage to a minimum while at the same time keep data organized in ways that make it is easy to find and extract information. Even though data doesn't always represent a conventional Spreadsheet, for want of a better word, we will often throughout this manual refer to these files as "Worksheets". WorkBook Worksheets are written compactly, extraneous cells used to dress up the file for listing aren't necessary because of the reporting utilities provided. This can conserve a considerable amount of memory in addition to making it faster to move about the file. When used as a Data Base Editor, each cell represents a field such as Zip code or company name. Since data entry is 30-50% of our costs and time, we want to keep the fields as general as possible that we may not later be restricted by lack of forethought. As you can see, the design of data and report formats is very important and should be a major emphasis when learning this system.

This manual is in several sections. "Getting Started" should be read before investigating the programs in this package. "Using the HP-71" is as much about a philosophy of using the computer in general as it is about using it with this package. There is a section on each of the programs with a discussion of their purpose, usage, and a list of commands. Part 5 explains the structure of WorkBook files and discusses methods for using the data in your Basic programs. Part 5 need only be read if you plan on writing your own programs. The inevitable Addendum is in the last section, listing some cautions and hints. A glossary is at the back of this manual.

Using the HP-71

As you already know, the HP-71 is a remarkable machine. It can address more memory than many desk top computers, yet it has a 4-bit processor and a smaller footprint than many pocket calculators. It speaks a very advanced dialect of BASIC, yet its programming methods are as different from its closest HP relative (the HP-75) as from the "standard" Microsoft BASIC. It is the multiple file structure and unlimited recursion which make packages like this one practical, and that is what this chapter is about. No attempt is made here to teach HP-71 BASIC, you may not even want to learn it.

The BASIC HP-71

Using the HP-71 as well as programming in both BASIC and Assembly Language are covered fully in a book written specifically for the 71 user. "The BASIC HP-71" contains numerous charts, tables and literally hundreds of examples. Much of the information of Vols I and II of the HP-71 IDS is included, but in more usable form. The book answers often-asked questions as well as offering new ways of using the HP-71. If you are new to the 71 or just wish to make better use of the computer, this book is to be recommended. "The BASIC HP-71" can be ordered through the dealer from which this package was purchased or directly from the author. Please write for ordering information.

Calling Programs and Calculator Variables

When you run a program it can access variables which you have used in CALC mode. This is an advantage because you can do some calculations (like using the statistical functions) then run a program which will perform further processing of those variables.

This is all well and good but, what happens when you run a large program with a considerable number of variables? It can alter your calculator variables without your knowing it until you go back to use them later. Also, when the program ends, it leaves behind a considerable number of variables in which you probably have no interest and wish weren't there wasting valuable memory. One solution is to use DESTROY ALL in the program, many do. Unfortunately you also loose your calculator variables.

There is an elegantly simple method to preserve your calculator variables for use in personal solutions and still be able to run variable-hungry programs. When a subprogram is CALLed the computer creates a separate environment, with its own variables. When the subprogram ends its environment is discarded and your calculator variables are left intact. Begin the program with:

```
10 CALL APROG @ SUB APROG
```

Of course, you can use any program name. The first statement tells the computer to call a subprogram which happens to begin

in the next statement. When the program ends what actually happens is the computer reclaims the memory used in the subprogram then goes back to read the next command following the CALL statement. The next statement is SUB which designates the beginning of a different program, hence the end of the current one, so the program terminates with an implied END. This is how the programs in this package operate.

Looking at RECURSION, at RECURSION, at RECURSION
It's all done with mirrors, actually pointers and return stacks. Calls to subroutines can be nested within themselves. In the same manner programs can call other programs and return to where they started. It is the second concept which is the most helpful in conjunction with this package. A program can call another program, even itself, then return upon completion! (this is the only sentence in this entire manual which ends with an exclamation point, note why). As an example, the WorkBook Editor program can call itself to edit another file (or another portion of the same file). Then, when you exit the second program, the original file will still be active with the cursor in the same place. The only catch is that the called program can't end with PAUSE because the computer will do just that. Of course, you can use **f-CONT** to continue, but the effect is less transparent.

HP-71 File Structure
There can be as many files in the computer as available memory will allow. File types can be BASIC programs, Text, LEX and others. CAT ALL can be used to list the available files. This listing can become quite long and cumbersome when looking for a particular file. The HP-71 has the ability to partition memory into :PORTS. There are many reasons for recommending the use of partitioned memory (such as not loosing files from a "Memory lost"). In the context of this package, it is helpful to place often-used LEX files and programs in a :PORT to keep the main catalog free for DATA files and temporarily loaded subprograms. There is also some conjecture that, if you have a great deal of memory, :PORT RAM causes less moving data about during running programs so the programs (usually those writing TEXT files) will probably run faster.

The WorkBook71 package consists of several separate programs, or "modules". Each module performs a specific kind of task such as data entry (the WB program) and printing reports (the REPORT program). The modules are as compact as possible and can operate independently with it being necessary only for the module you are currently using to be in the computer at any time. This is done to keep memory usage to a minimum. This method of working also makes it possible to choose only the applications you really need.

What's on the Disc

WBLEX	(Machine Language file)

The custom LEX file which must be loaded to use the programs in this system. This file extends BASIC with new keywords.

WB	(spreadsheet editor)

The main editor for WorkBook71 Data files. This is the Spreadsheet and Data Base data entry and review program. It is the largest module and is the workhorse of the system.

UTIL	(spreadsheet utilities)

Extension program for use with WorkBook Data files. Offers operations which are not part of the main editor.

TED, LED	(text editors)

Much of computing deals with Text files. These two programs are Text editors; TED is a Full Screen editor, LED is a simple line editor which is very compact and easy to use. These programs require other accessories for use.

PORTER	(file converter)

Performs file type conversions between Text and WorkBook Data files. Useful for exchanging data with other programs or other computers.

FINDER	(search program)

Input a series of qualifiers such as Zip codes within a certain range or city and FINDER will search WorkBook71 Data files for matching data. Outputs data which is compatible with Distribution Lists used by the REPORT program.

SORTWB (sorting program)

Used for alphabetizing fields in a WorkBook71 Data file. Especially useful when the file is used as a Data Base.

MENU (operating system shell)

A "shell" program which can be used to call other programs. When this package is run from MENU it will operate in minimal memory and with a minum of shuffling programs and files around.

PRINTWB (spreadsheet printer)

Prints a WorkBook71 Data file as a grid of columns and rows, much as it is displayed on a monitor, or lists it like a Data Base.

REPORT (text formatter)

The report printer module. Can be used as a Text Formatter with any Text file. Extensive command language allows interraction with WorkBook71 Data files as well as other operations like formula calculation, date, star graph plotting and extended printer control. Prints over 300 words per minute.

CHARSET (create alternate character set)

Enables the HP-71's alternate character sets. Two options are underlined and inverse (white on black). These characters are used throughout the system and may be used with your own programs using the WBLEX "HI$" function. This program is optional; it is not necessary to use this program or the alternate character set. Instructions for this little program are discussed in the Addendum.

WBSAMPLE

A completed WorkBook71 Data file set up for travel expense accounting with examples of formula and data usage. Can use the label file called "LABEL".

DBSAMPLE

A completed WorkBook71 Data file used as a mailing list Data Base.

TXSAMPLE

A Text file with examples of many of the REPORT commands. Use the REPORT program to print the file then edit the file with TED to see how it was written.

User Keyboard
These programs do not alter key assignments. Neither do they activate nor deactivate USER mode. When programs are waiting for a single keystroke such as 'Y or N', key redefinitions are ignored. When the flashing cursor is present and USER mode is set redefined keys are active. Several keys have special uses in the WorkBook Spreadsheet and Text Editors. It is best to either not redefine those keys or not have the USER keyboard active when running the program. These programs 'take over' those keys. Immediate execute key assignments are ignored in both programs. Nothing will happen when you press them.

Display Delay Rate
The WorkBook Editors ignore current delay settings. For other programs set DELAY to what you feel is a comfortable rate. Note that program execution pauses for the DELAY time when displaying data before moving on to the next instruction.

The RUN key
You can use programs in this series by using either CALL filename or RUN filename. Using RUN has an advantage in that the HP-71 will run the most recently-run program again by pressing the RUN key. If, for instance, you often use just the WorkBook Editor enter RUN WB. The next time you want to use the program you can do so by pressing the RUN key again. If the program being RUN is in RAM be sure it is SECUREd to prevent accidental alteration after the program is thorugh running, it will be the current file.

10-minute Timeout
Normally the HP-71 will timeout after waiting aproximately 10 minutes for a keystroke. The two WorkBook editors will not timeout. Instead, after 10 minutes of inactivity they will enter Command level. Other programs in the package will timeout normally.

Power Consumption
While running BASIC programs causes an increase in battery consumption, the programs in this package were designed to minimize power waste. In Browse level or when the flashing cursor is present relatively little power is consumed.

When the HP-IL loop is active, power consumption is somewhat increased. Maintaining a monitor increases power usage again. It is recommended that an AC adapter be used when using an external keyboard.

Editing Programs
Most of the programs in this group are editable. That is, you may make them the current edit file and examine the code. Many lines in the programs are longer than 96 characters and

are therefore uneditable. If you edit one of these long lines (of a RAM-based program) and press **ENDLINE** either one of two things will happen: The computer will issue a syntax error because of missing code, Or the part of the line that was in the display will be accepted and replace the complete line in the program. If the latter occurs then the code on the end of that line is lost and cannot be recovered. The loss of even a single instruction can make a program unusuable. For this reason it is recommended that you do not attempt to edit these long lines.

Making Backup Copies
Most of the program and data files used by this system are copyable. It is recommended that before even using the system you make a spare copy of each file on the Disc or Cassette to another medium. Discs and Tapes are magnetic media and subject to data alteration by exposure to anything from a telephone to a cup of coffee.

The WorkBook file
This system uses a DATA file with fixed number of records and record length. The first two records are used by the system to store data about the file such as last cell edited or number of columns in the file. This data is read whenever a file is assigned, then updated when through editing the file. The data is in a compacted format and its use is generally transparent. Each subsequent record in the file corresponds to a cell. Data is stored in sequence beginning with the cell in the upper left corner. The last row in one column immediately preceeds the first row in the next column to the right. The chapter 'Using WorkBook Files with your programs' has a more in depth discussion of file structure.

Using HP-IL
The WorkBook system will support a printer, mass storage device and a display device. Printer and display devices must be assigned before running the programs, the programs will not assign them. For this reason you may designate any device that can process the data as the device. This is helpful if you are using, for instance, a non-HP printer with parallel interface or perhaps an RS-232c display device or keyboard. Assigning devices is discussed in the HP-IL Interface Owner's Manual.

The WorkBook Editors checks to see if there is an active display device assigned when they start. If there is one then the programs will maintain a display of cols and rows. Prompts for input which are normally shown only on the LCD are also shown on the bottom rows of the display device. The Text Editor offers direct vertical cursor movement. Other programs will support a display device but do not necessarily provide a formatted display.

Running a program which requires a printer for output, such as PRINTWB, without a printer will cause all printer

output to be sent to the LCD.

The first device found on the HP-IL with an accessory ID from 16 through 31 will be used as the mass storage device. This corresponds to the Hewlett-Packard 82161A Cassette Drive and Hewlett-Packard 9114A and 9114B Disc Drives. Other devices with the appropriate accessory ID and which have a similar response to high level HP-IL commands should also be usable.

The usual method for a program to determine the existence of mass storage from a running program is to poll devices until one is found. This is effective unless the loop is inactive in which case the computer will wait for several seconds for a nonexistant device to respond. Instead of requiring this wait occasionally a program will ask if you would like to use mass storage, even if it is not present.

Required LEX File

A custom LEX file called WBLEX must be present in RAM, either :MAIN or in a :PORT, or in an EPROM. The purpose of WBLEX is to extend the input capabilities, data formatting and to speed formula recalculation. The programs will not function properly without this LEX file, and they will not load it if it is not found in memory. You should always be sure that the WBLEX LEX file is in the computer before running the programs.

Memory Requirements

Be sure you have enough memory free to accomodate the desired program and associated data files. Each program requires from 600 to 1100 bytes of variables as they operate. Programs called from the WorkBook Editor will require additional memory. If you have a limited amount of memory it is probably best to have only one major application in memory at a time. This can be accomplished by running the system from the MENU program. In that case each program will be loaded from Disc as needed then deleted from RAM when finished.

Introduction

The WorkBook Editor is a general-purpose program which uses a specially formatted DATA file as a Worksheet. If you have not done so, it is suggested that you read the chapter 'Using WorkBook71' for an overview before editing files with this program.

Run the Program

The program file is called 'WB'. It may be run from the MENU program or run directly from the keyboard. It may either be RUN or CALLed. It requires no parameters. Be sure you have enough memory for the aproximately 1000 bytes of memory the program uses while it runs in addition to your other files. WB uses the **ON** key so that key cannot be used to suspend the program once the WorkBook file is being edited. Use Command level **E** (discussed later) to exit this program.

At the first prompt the program will ask you for a file name. Enter the name of a valid WB DATA file or a new file name. Do not include a mass storage device name with the file name.

```
WB71 file:
```

If you press **ENDLINE** without specifying any file name the program will do a CAT ALL. Terminate the CAT ALL by beginning to type the file name. At this point you may suspend WB by pressing the **ATTN** key.

If the file is of the wrong type or there is insufficient memory to continue, the prompt will return. If the specified is a valid HP-71 file name and the file is not currently in RAM you will be presented with the following options:

```
New,Load,Virt
```

Press **N, L** or **V.** at this prompt. It is not necessary to press **ENDLINE.** New means create a new file. Load means look for the file on the first mass storage device and load it to main RAM. Virt means use a mass-storage based (or "virtual memory") file of the name previously specified.

When a new file is created you have the option of specifying three parameters for the file:

```
col,row,len: 8,8,40
```

This specifies the physical size and format for the file. The three requirements are number of columns, rows and individual cell capacity (length). All three parameters are required and must be separated by commas. Mathematical expressions may be used. While the number of cols and rows may be altered later, the individual cell length may not. The default is a file of 8 cols by 8 rows with a cell size of 40 characters. Once the

file is assigned you are editing the file.

Setting up the File

The WorkBook DATA file is the backbone of the system. When you first create one you are designing the format to be followed throughout its use. Several thoughts should immediately come to mind:

- How much data do I think will be needed?
- Am I sure my needs later on will be accomodated by the selected size?
- How much memory is available?
- What kind of work am I going to do with the file?
- If I am going to do numerical analysis how complex (long) will formulas be?
- If I am going to use the file for tabular or Data Base data what is the maximum field length I will need?
- What physical shape will the file need? For instance, if you are writing a weekly bookkeeping spreadsheet which will cover one month, you will probably need columns for each week and a total column, and quite a number of rows for such things as unpaid invoices or closing inventory. This file would probably be 6 or 7 cols by perhaps 30 rows, and since formulas are straight forward, cell length of perhaps 40 characters. Another file might contain names and addresses in a mailing list. At most there will be fields, so we need about 7 rows. The number of cols is the total number of entries you will be making, say 100 people. There will be no formulas, and names are fairly short so a cell length of 25 characters is probably enough.

If the file is to be used as a notebook some approximation of use should be made. When using a monitor, the most comfortable cell length is probably the width of the display device minus 5 (for the Row label). Used handheld, it is a matter of personal preference. Some prefer to enter entire sentences on one line, others prefer to be able to see most of the Text when they back through the file. This can only be done with short lines.

Once the file is created, the PORTER program contains utilities to move data to either a larger or smaller file. You should consider using PORTER as a last resort. There is no substitute for organization and planning. Extra cols or rows may be added as the program runs, but they will be added at the end of the file, and it takes several seconds.

Conserving Memory

The Text Formatter program can be used to embelish the printed format of any Worksheet. Extra blank or "dressing" cells can therefore be eliminated. Another advantage of keeping the file compact is that it is faster to move between actual entries without having to move past multitudes of blank cells.

Displayed column width has no bearing on memory usage but

the cell width, that is, the maximum number of characters which may be stored in a cell, is very critical. By keeping the numerical display format (FIX) to a minimum number of decimal places you will again be able to keep required cell width to a minimum. Of course, FIX9 will use seven more characters than FIX2.

Figuring Memory Usage

Since cells are of a fixed size it is easy to determine the memory files will consume. Memory required is the line length plus 4 multiplied by the total number of cells plus 2.

```
(LINE LENGTH+4)*(#COLS*#ROWS+2)
```

Using a Monitor

The WorkBook Editor provides a formatted display if a Video Interface is available. The file is displayed in a grid of columns and rows on the top part of the screen. The current edit cell is displayed on both the bottom of the screen and on the computer's built-in LCD display.

```
        1     2     3     4     5
1
2     1.80
3
4
5           3.60
6
7
8

 2,5 +3.60 @[1,2]*2
```

In the example above, the current edit cell is displayed on the bottom of the screen. The line begins with a Col (2) then Row (5) label refrence. As you can see, cell [2,5] on the top of the display contains the value 3.60, the same as in the cell. There is a formula in the cell (formulas are discussed later) referring to cell [1,2] which contains the value 1.80 (and may very well also contain a formula, but formulas are only visible on the edit line). The contents of each cell is displayed on the Spreadsheet grid with as many characters as will fit within the column width, but the entire cell contents are displayed on the edit line.

Each time you move to another cell its contents are shown on the edit line. In this example only, Col 1 through 5 are shown. Should you move to Col 6 the entire display will shift over to include Col 6, Col 1 will not be shown. Rebuilding the display will take a few seconds. You can view a cell which is not in the display by using **g-ON** without having to wait for the display to change to a new screen.

The program is provided configured for a 32-Col by Row display (or no display device at all). The Addendum gives

instructions for changing to other display configurations.

Using Virtual Files

Memory constraints become much less of a problem by the use of virtual files. By virtual we mean a file which is resident on Disc or Cassette with only a small portion being kept in RAM at one time. If the file is kept to a reasonable size it may, at different times, be used either from RAM or Disc. The obvious advantages of Virtual Files are that files as large as the capacity of the Disc may be used, or a file that you would usually use from RAM may be used even if you suddenly find yourself with a computer filled to the brim with data you don't want to delete.

There are disadvantages to using Virtual Files. A "worst case" test file recalculated in RAM took 27 seconds. The same file resident on the HP 9114A Disc took 36 seconds. The very same file running on the HP 82161A Cassette Drive took 130 seconds. The file had a formula in each cell refer to the previous cell. Fewer refrences to other cells would have narrowed the speed gap. It is obvious that, especially with Cassette, only the portion of the file that needs updating should be recalculated. During recalculation each cell is examined to determine if it has a formula, even if it is an empty cell. Of course, if the cell doesn't have a formula then it is ignored, however it does require some time to make this determination.

Formulas with refrences to other cells take a special importance with Virtual Files. The file is stored on Cassette by cols. That is, the last row on Col 1 immediately preceeds the first Row in Col 2. For this reason formulas refering to cells to the left or right of themselves will calculate somewhat slower than if the refrence had been to a cell above or below.

Mass Storage Warnings

The HP9114A Disc Drive should be used with caution with Virtual Files. This one scenario is not meant to discourage the use of either the Disc or Virtual files, it is mentioned to make you aware of a potential problem. The HP9114B will not be as likely to experience the following.

Files on the 9114A run nearly as fast as in RAM. For this reason it is easy to forget how hard the Disc is working. The 9114A's battery is rated at 8 hours of intermittent use or about 40 minutes of continuous use. It is quite possible that, by constantly moving around the file or recalculating several times, the recharger won't be able to keep up with the drain. The flashing red fault light comes on to warn us of the problem but, it will probably have been on for some time before being noticed. When you exit the program, which you will probably try to do when the Disc faults, WB tries to save current status to the beginning of the file. The Disc may not allow this operation, in which case the program won't be able to terminate properly. If possible, leave the computer and

Disc Drive idle until the battery has had a chance to catch up, then try to exit again. As a last resort INIT 1 (ON-/) can be used to stop the program. The problem with this method is that it leaves the channel associated with the Disc file assigned. Again don't turn off the computer or Disc. When the Disc seems to be up to it enter END ALL to close the file. This will not update the status of the file, but will do an orderly exit. It should be noted that sometimes the charger can keep up as long as the Disc is continously running, the continual start-stop takes the toll.

Formulas, Recalculation

The three types of cells are Text, Numerical, and Numerical/Formula. The latter two have in common that you may enter formulas which WB will compute, then return the results to that cell. Numerical cells contain only the results of that formula. Formula cells contain both the results and the formula itself. This type of cell will be updated when you recalculate the Worksheet.

The three cell types are differentiated by the leading character:

```
First Char '='  A Numerical (results only) cell.
First Char '+' A Numerical/Formula cell.
```

Any other first character designates a Text cell. The examples in this section assume FIX 2 which is the display format a new file will have, and that you are in Edit level. If you are running the program now and you see FLAG 1 lit (next to the PRGM annunciator) press **g-ENDLINE** to enter Edit level. Edit level is discussed later in the command section.

One advantage to this system is the ease of changing from one cell type to another. For example, a formula may be temporarily changed into Text by replacing the + with perhaps a space.

Numerical Cells

This cell type may contain a number which is formatted to the specified FIX setting. To enter a Numercal cell first enter an = then either the number or a formula which will result in a single result. The formula will be computed then the equals sign and results of the formula are entered into the cell with as many decimal places as specified by the current FIX format. Enter:

```
=7*8            Now press RUN
=56.00          (assuming FIX 2)
```

Formula/Results Cells

There is another numerical cell type available. A Formula/Results cell (usually referred to as just a Formula cell) is entered in the same manner. However, the formula is retained in the cell.

```
+7*8
+56.00 @7*8
```

When you are using a monitor only the number will be displayed in the cell coordinate. However, the formula is still available on the edit line.

You've changed your mind, The cell should be 7*7. Move the cursor over the 8 and press 7 then **RUN.**

```
+56.00 @7*7
+49.00 @7*7
```

Any time you change any part of the cell it is recomputed when you press **RUN** or move to another cell. To recompute a cell without changing the formula overwrite part of the number with, for instance, spaces. Either Numerical or Formula cells will only calculate a formula following an @. Even if a cell has been calculated by pressing **RUN**, it will again be calculated when you exit the cell if the cell contents have changed from what they were when you first began editing that cell.

Program Variables

Your own calculator variables are not available when using WorkBook, in fact, they are ignored. A separate set of variables used by the program may be useful in some formulas.

```
C  - Current edit column
R  - Current edit row
C9 - Last Col in this file
R9 - Last Row in this file
D  - Location of display device
G  - Global (whole file) display format
L  - Max number of characters in cell
```

Cell Refrences

In addition to simple formulas and the variables above, cells may refrence other cells. Cells are refrenced by their col and row number. When used in formulas enter a bracket ([) then col number, a comma (,), then row number and, finally a closing bracket (]).

```
+2*[2,4]
```

This formula says take the value in cell [2,4], the second column, fourth row, and multiply it by two and return the result. Brackets are used like parentheses except that they are used ONLY to delimit cell refrences. They may be used without extra parentheses to separate functions.

```
+SQR[2,4]*(7+[1,3])
```

Mathematical Functions

Standard HP-71 or LEX file functions which requires only numbers and returns a single numerical result should operate satisfactorily. The restrictions on functions (such as those in the Math ROM) in regards to their use in CALC mode also apply in formulas in cells. Unlike CALC mode, simple string functions are allowed.

Formulas refrencing other cells recall the value of that cell to the FIX precision of the number in the cell whether it is a Numerical or Formula cell. Refrences to Text or empty cells return zero. Mathematical exception flags are not altered by this program.

Spaces within formulas are not allowed. Remarks are allowed at the end of a formula if preceeded by an exclamation point (!). Improperly-entered formulas will not necessarily display an error; instead, as much of the formula as makes sense to the computer will be interpreted. This may or may not provide useful data. So, spelling-in formulas should be on a higher level than spelling exhibited in this manual.

Flags

The WorkBook71 editor uses flags 0-1. Any other user or system flags may be used in formulas. WB also stores the status of flags 2-3 with the file, and they may be used in formulas. Other flags are not used by this program.

SUM Function

The sum function (called FNS) is used in formulas to look up the value in a range of cells and add them together, returning their sum. The function is used when you need to sum more cells than can be entered into the maximum cell length of the cell.

Specify a starting coordinate and ending coordinate. All of the cells between those two points will be read, their values summed, and the results returned.

```
+FNS([3,1],[3,7])
```

In the example, each cell between [3,1] and [3,7] will be read. Any range of cells may be specified. As with recalculation, cells are read by Cols. The range may extend to two or more Cols as desired. FNS may be used only once in a formula; if it is used more than once in a formula then the results of the first usage will be repeated.

User Labels

Creating User Col and Row labels is described in Command Level discussion. Assuming that they exist, formulas may refer to either the Label or the col/row number. It is not necessary to use the entire Label, just enough to insure that it is not confused with a similar but earlier label in the file. We will call a Col Label "Total" and a Row Label "Week".

```
2*[tot,w]
```

When the formula is computed the default Col and Row Labels replace the USER Labels in the formula. This is done for faster recalculation and so that the formula will still be valid if the User Labels are not available later.

Recalculating the WorkBook File

Numerical/Formula cells may have refrences to other cells. Change the refrenced cell and the first cell no longer contains up-to-date data. This happens each time you change a cell which is refrenced by another cell. Periodically you may want to be sure that all cells are current. Another reason for wanting to update the Worksheet is to do "what if..." with the file. For instance, change [units,sold] and then view how it affects [net,profit]. To do this we recalculate the Worksheet. In any level press **f-CALC** and the display will show:

```
calc from:1,1
```

This refers to the beginning cell to calculate from. The default is the upper left corner or 1,1. You may change to any other cell in the file using either default or user labels, then press **ENDLINE**. Brackets are not necessary but both Col and Row must be specified. You may specify USER labels for Col and Row refrences. Then next prompt is:

```
to:8,8
```

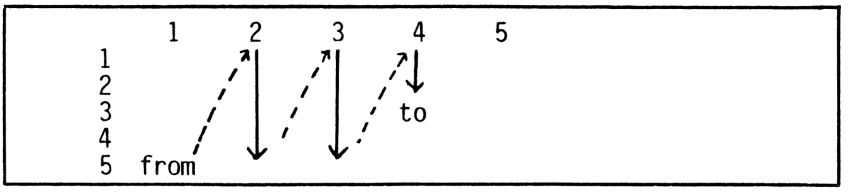
The default is the lower right corner in the file. Again you may change it or leave this value. When you press **ENDLINE** this time the display will change to:

```
working
```

This will remain in the display while the file is recalculated. Recalculation will take from a few seconds to several minutes depending on the complexity, size and if the file is in RAM or on mass storage.

Recalculation Range

The two prompts asking for first then last cell to compute are inclusive. That is, any cells between those two extremes is calculated.



In the example above, if you had specified [1,5] as the starting cell and [4,3] as the last cell then all cells between those two will be calculated. This includes [2,1], [3,1] and so forth.

Order of Calculation

The physical WorkBook file is organized with one Col following the previous one. Col 2 physically follows Col 1 in the file. To minimize the time needed, recalculation is always done by Cols. After the last Row in one Col is calculated the first Row in the next Col to the right is calculated.

How Recalculation Works

During recalculation each cell is read. If it is not a formula then it is ignored and the program goes on to the next cell. When Numerical/Formula cells are encountered the data following the @ is read. Each cell refrence is looked up. Again text or blank cells are read as zero and numerical cells are interpreted to the precision of their Fix setting. Once cell refrences are replaced with the actual values the formula is computed. Finally, the cell is rebuilt with number, @ delimiter, and the original formula (with cell refrences). The program then moves to the next cell down in the same Col, or, if it is already at the last Row, will start at the first Row in the next Col.

Circular Refrences

There is a formula in cell [2,2] in our hypothetical Worksheet which refers to cell [3,4]. Cell [3,4] has another formula, this one referring to the value in cell [2,2]. When we recalculate the file cell [2,2] refrences [3,4] and is changed accordingly. A moment later the program comes upon cell [3,4] which itself refers to [2,2] so it reads that cell and updates [3,4].

As you can see, each time the Worksheet is recalculated the two cells will change. There can be no answer to these formulas except an eventual overflow or underflow error. This is called a circular refrence, and it isn't always as obvious as in this example. Any number of cells may be involved.

WorkBook71 operates in three main levels (or modes). This program has a different implementation of the three levels than many spreadsheet programs to better take advantage of handheld use and flexible data input. In Edit level the current cell is displayed with the flashing cursor present. The entire keyboard is active and you may change the cell contents.

Whenever the program displays a prompt and asks for a decision there are two different forms it may take. When the cursor is present you may alter the displayed line as needed, then press **ENDLINE** to signify that you have completed data entry. When the cursor is not on and you are presented a prompt then only the uppercase characters (and alternate character set, if it is active) are valid inputs, and it is not necessary to press **ENDLINE**. In either case you may cancel with the **RUN** key or cancel and return to Command level with the **ON** key.

Browse Level

Pressing **g-CMDS** toggles between Edit and Browse levels. Browse level (called "Top level" on many spreadsheets) is used for looking about without the possibility of accidently altering a cell by a misplaced keystroke. You can tell that you are in Browse level because the annunciator for FLAG 1 is lit and the flashing cursor is not present, though the cell is still in the display. In this level only the bottom row black keys and the f-shifted function keys are active. You may also find moving from cell to cell somewhat faster than in Edit.

To edit the current cell while still in Browse press the **RUN** key, Browse returns when you exit the current cell.

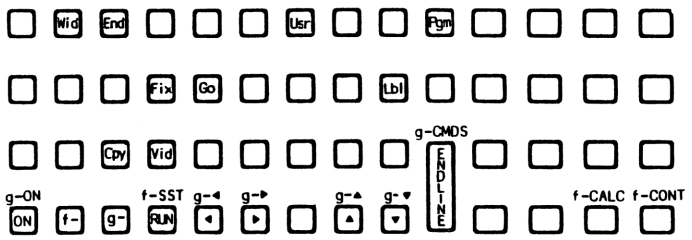
Command Level

The third operating mode is called Command level. Enter Command level from Edit or Browse by pressing **ON.** Occasionally the program will enter this level if an error has occured (such as interruption of continuity on the HP-IL or not enough memory for the task at hand). The Command prompt shows some of the current file status and available memory. The Browse Level keys are still active; additionally several other keys are active which allow you to perform functions other than editing cells or looking about (which are done in the other levels). The Command level prompt displays a one-to three character abreviation of the available functions and is on two lines. The number at the right end of the first prompt is currently free RAM memory. The second Command prompt may be displayed by pressing a non-command key (such as any of the unshifted white keys) and will remain in the display as long as the key is held down; release the key to return to the first Command prompt. The two numbers on the end of the second Command prompt represent the number of cols and number of rows in the current file. Command Level functions require

a single keystroke to activate. It is not necessary to press ENDLINE to enter them.

The way it works is that you enter Command level, perform your task, and when it is completed Browse or Edit level (which ever had been active) returns. When you are offered a keystroke from among several, such as 'Global/Cell' you may cancel the operation and return to Command level by pressing ON or return to Edit/Browse level by pressing **RUN.**

WorkBook71 Keystrokes



EDIT LEVEL

The entire keyboard is active for data input and review. In addition, several keystrokes perform special operations. Each of these keystrokes will terminate input in the current cell and, if you have changed the cell from the last time you edited the it, will recalculate the contents of that cell. Note that the entire file is only recalculated when you specificially command it. The following keystrokes are active in Edit, Browse and and Command levels.

ON

Press this key to enter Command level. Also used to cancel an operation (except data entry in the current cell) and return to Command level.

RUN

Terminate entry in the current cell, calculates any formula. The cursor does not move from the current cell. If you are currently in Browse level will allow you to edit the current cell. Note that the cell will be recalculated when you exit the cell if it had been changed. It is not necessary to press **RUN** to terminate input.

UP ARROW

Terminate input as with **RUN** but moves up one row in the current col.

DOWN ARROW

Terminate input and move down one row in the current col. If you are at the bottom row will stay there.

g-UP ARROW

Terminate entry and move up to the first row in the current col. If you are currently at the first row in the col then it will stay there.

g-DOWN ARROW

Terminate entry and move down to the last row in the current col.

ENDLINE

Terminates input in the current cell as with **DN ARROW** but, if you are at the bottom cell in the current col, will move to the first Row in the next col to the right. This is helpful for continuous data input such as filling-in names in a Data Base.

f-CALC

Terminates input, but also recalculates the other cells in the file which have formulas. The first prompt asks you the cell from which you would like to begin recalculation. The default is cell 1,1. Press **ENDLINE** without changing the prompt to begin recalculation from cell 1,1. Cell coordinates may be specified with either USER or default labels.

```
calc from: 1,1
```

The second prompt asks for the last cell you would like to recalculate. The default is the cell in the lower right corner of the WorkBook file.

```
to: 8,8
```

To recalculate the entire just press **ENDLINE** at both prompts. The cells are recalculated sequentially so that if you specify 1,4 as the first cell and 3,3 as the last cell, all cells between them, including, for instance, 2,1 will be recalculated.

Do not interrupt the computer while "working" is displayed. Larger and more complex files take longer to recalculate. Assume a few seconds for a simple file up to several minutes for a very large file, a virtual file, or one with very complex formulas. If you are using a large file residing on cassette with cell refrences spread throughout the file it is very highly recommended that you only recalculate

the part of the file that you really need to or the life expectancy and reliability of the tape may be seriously affected. You may not find this to be a problem with Disc-based files.

Because of the structure of the files, to keep recalculation as fast as possible, recalculation is by columns. Recalculation progresses from the first cell specified to the bottom of that col then from the beginning of the next col to the right and so on.

The problem with specifying too short of a line length for the job at hand is that if the resulting contents of a cell are longer than the allowed line length the recalculation will halt and the long line will become the current cell for you correct. After correction you will have to again press **f-CALC.**

f-CONT

Rebuild the display and continue from where you are. This is helpful if you are using a display device and, through unseen forces, the display device has become an inaccurate representation of the file. It clears the display then completely rebuilds the display on the monitor. This keystroke acts as a terminator key if you are not using a display device.

g-ON

Display contents of a cell without actually moving the cursor there. While the requested cell is in the display, you are effectively in Browse level with the current cell (not the cell you are viewing) as your starting point. Terminate the display of this alternate cell by pressing an edit key. For instance, press **RUN** and the current cell will return to the display. Please do not confuse the use of **g-ON** with the use of **f-VIEW** which is used to view key redefinitions or **f-SST** which is used to search for a series of characters.

g-CMDS

Toggle between Edit and Browse level. That is, if you are currently in Browse, will change to Edit; the reverse is also true. The FLAG 1 annunciator is lit to indicate that you are in Browse level and cleared to indicate Edit level.

f-SST

Search for a string or number and, if it is found, make that the current cell. It is easy to remember the action of this key by remembering that SST stands for Search for STring. This function is especially useful in a DataBase file to find, for instance, a phone number when all you can remember is a last name. The first prompt is:

Abs/Rel

Abs is short for absolute and means to search from the beginning of the file. Rel is short for relitive and pressing **R** means that you would like to search from your current position in the WorkBook file. The second prompt is:

Col/Row

Pressing **C** specifies search by columns, **R** specifies by rows. At the third prompt enter the string which you would like to find, then press **ENDLINE.**

find:

The search can take from a second or two to quite some time depending on if the file is in RAM or on Disc and where the string is relative to your starting position.

BROWSE-LEVEL KEYS

In addition to the EDIT level keys discussed above, several other keys are enabled in Browse level. Line editing arrow keys are used while editing cells but they are not needed in Browse so they become available to move between columns. **RUN** is used in Browse to edit the current cell. When you exit the cell Browse level is again in effect. These keys are also active in Command level.

LEFT ARROW

Move to the column to the immediate left of the current col while remaining in the current row.

RIGHT ARROW

Move to the next col to the right while remaining in the current row.

g-LEFT ARROW

Move to col number 1.

g-RIGHT ARROW

Move to the rightmost col in the file while staying in the current row.

g-UP ARROW

Move to row number 1 while remaining in the same col.

g-DOWN ARROW

Move to the last row in the file.

In addition to the Browse level keystrokes, several other keys become active in Command level. These single letter keystrokes represent the first letter in the command name. For example, **P** to call a Program.

The command level prompt is split up onto two lines, to view the second line press and hold down a non-Command level key, such as the un-shifted white keys, for as long as you wish to view the prompt. The first prompt shows three of the Command level functions and currently free main RAM memory. The second Command prompt shows the remainder of the functions plus the total number of cols then rows in the current file.

Besides enabling these extra functions, Command level makes it easy to temporarily activate the additional arrow keys which have other uses in Edit level. Press **ON** then an arrow key to move to another col.

C	(copy a cell to a range)

This function is used to copy a cell to a range of cells. As with recalculation, Copy is column oriented and works sequentially on all cells between the specified range.

At the first prompt specify the source cell. The second prompt is "start:"; enter the first cell coordinate in the range. The final prompt, "end:", is used to specify the end of the range. The source cell, including cell format, will be copied to each cell in the specified range. The cells will not be recalculated, nor will formulas with cell refrences be adjusted.

This function is useful when a similar formula or standard piece of information will be needed in several locations in the file.

E	(exit the program)

Exit the program. At this point the current status of the file (cursor position, label file) is updated, user flags and system status are restored, then the program ends. If there had been a display device active the final screen will remain.

F	(number fix format)

Set the format used to display numbers. The first prompt is 'Global/Cell'. Global refers to the setting for the whole file (that is, cells without a local format). Cell refers to the format of the current edit cell. Press either **G** or **C** to change either the fix setting for the whole file, or the current cell.

When a new WorkBook file is created the Fix for the file (global Fix) is set to Fix 2 and each cell is set to Global setting. That is, each cell will display numbers to the precision specified by the global Fix. At the second prompt

the fix may be set for values of 0-9 and G. A cell set to 'G' assumes the Global Fix precision.

G	(goto)

Goto a cell. The prompt begins with the current cell. Change it to any valid cell coordinate in the file. You may specify either default or user labels or a combination of both. As when used in formulas, separate the col and row label coordinates with a comma, and user labels may be entered using just the first character or two, (enough to insure the name is not confused with another label).

Note the difference between Goto and View (activated with the **g-ON** key) functions. Goto will make the specified cell the current edit cell while View will not move from the current cell.

L	(assign label file)

Establish or change the assigned Col/Row Label file. To keep the size of the WorkBook editor as small as possible, the UTIL program is used to write or change Label file contents. The label file is a separate Text file and may be used by several different WorkBook files. See the section on the UTIL program for requirements and instructions for use of Label files.

P	(call a program)

Calls a sub-program without passing any parameters. If the called program is not in RAM or ROM it will be searched for on the first or only Disc. If the name is not valid or the program is not found, this program will continue from where it was left. If the called program ends with an END statement then this program will continue exactly where it was left. To continue this program after using a program that stopped with an error or PAUSE enter END then CONT.

U	(toggle user labels on/off)

Toggles between displaying with User or default Labels. You can tell which is specified by the status of the FLAG 0 anunciator. If FLAG 0 is set then the user Labels, if available, are active.

V	(toggle Video on/off)

Toggles Video display updating on/off. The cell is still displayed on the bottom of the display but cell changes and movement do not change the grid as shown on the top lines of the display. This is helpful for viewing one section of the file while editing another. Has virtually no effect when no display is used. This status is not saved with the file.

W (set col width)

Set col width. Default is 8 and may be set between 1 and the width of the display device. Designates the width of each col to be displayed on a monitor. Does not affect the number of characters which may be stored in each cell or the display on the built-in LCD. The col width may be changed when printing a file, so this setting is primarily important for use with a monitor.

The program maintains a record of the current cell coordinate as well as the range of cells which will fit on a display device, even if the monitor is not currently active. This may occasionally cause a slight pause as coordinates are updated. Setting the col width to 1 when not using a monitor will cause this pause to occur less frequently.

Much of data entry on computers is in the form of Text files. HP provides a Text Editor in the HP-41 Translator, FORTH/Assembly and Text Editor ROMs. The HP program is complete, though often confusing to use and somewhat slow. We have included an alternate form of Text entry in WorkBook71 in the programs "TED" and "LED"; these programs have a command structure similar to the WorkBook spreadsheet editor (WB) and are easy to use, though lack some of the features of the HP program in order to save some memory.

TED (short for Text EDitor) is designed to be used primarily with a video interface and monitor. LED (for Line EDitor) can be used with a monitor, though it has as little support for it as the HP editor. LED is for use in the field; it is compact, fairly fast and uses less memory for both the program and its environment than the HP editor uses just for the environment. LED does not support the monitor functions of TED, but is otherwise similar.

Files must be in :MAIN or :PORT RAM. Virtual files may not be used. A major design goal of TED was full-screen editing and for that reason it operates in blocks of screens - as many lines as will fit on one monitor screen regardless of if there is a monitor connected. LED works a line at a time and does not try to fill the screen.

The programs are always in "replace" mode when entering data. That is, data is always entered after the last data in the file or it replaces the data being displayed. While the INPUT function is not used in the programs, the effect is much the same as a series of INPUTs.

The two operational levels (or "modes") are Edit and Command. Edit level is used for data entry and review and there are only a few control keys. Command level is used to gain access to other functions. For example, to add new lines between lines use the command level function "Ins" to add blank lines. Since the minimum file for TED is always one screen full, and several operations can leave extra blank lines, the command-level "Bln" is used to eliminate these blank lines. "Del" is used to delete a series of lines regardless of if they contain data.

Two numbers are always displayed on the bottom left of the screen when a monitor is being used with TED. The first number is the record number of the current edit line. Since Text file records begin with zero, line one is actually record zero, we will refer to actual record number to avoid ambiguity. The second number displayed is currently-available memory.

It is not necessary to press **ENDLINE** to enter the current line. Moving the cursor up or down enters the data from the current line into the file. Unlike EDTEXT, you may hold down an arrow key or **ENDLINE** to move continuously through the file.

Getting Started
Before RUNning or CALLing the program, be sure you have WBLEX and any one of the following programs in your computer. It will not run without one the these:

FORTH/ASSEMBLER ROM, HP-41 TRANSLATOR ROM, TEXT EDITOR ROM
TEXTUTIL LEX File #03183-71-8 HP-71 Users' Library

As furnished, TED is configured for a 32 column by row display device. The proceedure for customizing the WorkBook Editor, as discussed in the addendum, also applies to TED. It is not necessary to configure TED for the maximum size the interface can display. Of course, LED does not need to be configured.

TED is also useful without a display device, in which case the parameters set are unimportant. When using TED without a monitor it will still pause occasionally to "change screens". LED does not stop to change screens.

Using the Text Editors
RUN or CALL the program. The first prompt will ask for a file name; enter a new file name or the name of a TEXT file in RAM or on Disc (LED will not load files) then press **ENDLINE.** If the file is not in RAM the computer will ask if you would like to create a new one or load it from the first (or only) mass storage device.

"New/Load" -or- "New?"

Any error such as not enough memory or wrong file type will cause the original file name prompt to return.

When you have assigned the Text file to edit, the screen will show the first portion of that file. If it is a new file, the screen will be blank except for the the flashing cursor in the upper left corner and the line number and free memory in the lower left. Of course, if no monitor is connected, the LCD will be blank except for the flashing cursor. To begin using the program just start typing. There is no "wraparound", that is, a line is a line. When you reach the end of the line press a terminator key (discussed next) to enter the line and move to the next line. Some conservation should be observed when filling a line with Text. If the line is wider than the display width then you could end up with a nearly unreadable display whenever you move to a new line on the monitor, though no data will be lost.

Arrow keys can take you up or down a single line or a full screen. **f-SST** will find the first or next occurance of a word and place that line at the top of the screen and the cursor on the first character of the word. Command level is entered by pressing **ATTN.** The Command-Level "Go" function can be used to move to a specified line number.

EDIT-LEVEL KEYS

ON (enter command level)

Terminates input in the current line then enters Command Level. May also be used to cancel an operation (except data entry) and return to command level.

ENDLINE

Terminates input on the current line. Moves the cursor to the beginning of the next line. If the line had been at the bottom of the screen then TED will move to the next screen. If the line had been at the end of the file then a new screen of blanks will be created. Only a single blank line will be added by LED.

RUN

Terminate input and remain on the same line. Also used to exit Command Level and return to the current edit line.

DOWN ARROW (move down one line)

Terminates input on the current line then moves to the next line in the file. If at the bottom line in the screen will move to the next screen. Note that the cursor will not move farther right than the last character on the line when moving up or down.

g-DOWN ARROW (move down one screen)

Terminate input and move to the top of the next screen without homing the cursor.

UP ARROW (move up one line)

Terminates input on the current line then moves up one line without homing the cursor. If you are at the top of the screen will move to the previous screen.

g-UP ARROW (move up one screen)

Terminates input on the current line then moves to the top of the previous screen. LED moves to the beginning of the file.

f-CONT (rebuild screen)

Terminate input on the current line then rebuilds the display. Useful if the display has scrolled inadvertently and you wish to eliminate the jibberish. Causes a short delay even if used without a monitor. Not supported by the LED program.

f-SST	(search for a string)

Search for a string within the file and, if it is found, place that line at the top of the screen with the cursor on the first character of the string. Comparisons are made without concern for case so that "BEEP" and "beep" both will be interpreted the same.

The first prompt is "Abs/Rel". Abs means to search from the absolute beginning of the file regardless of where you currently are. Rel means to search starting with the line following the current line.

COMMAND LEVEL KEYS

The Command prompt is displayed on two lines. View the second line by pressing a non-command key, such as an un-shifted white key. The line will remain in the display for as long as you hold down the key; releasing the key returns to the first Command prompt. The number displayed on the first line is currently-available memory. The numbers on the second prompt are first the record number of the current edit line followed by the number of the last record in the file.

Command-Level functions are activated by pressing the key corresponding to the first letter of the name of the function, for example **I** for "Insert"; is not necessary to press **ENDLINE.** Edit-level control keys (such as the arrow keys) are also active in Command Level. Pressing **ON** will cancel an operation and places you in Command Level. **RUN** cancels operations and returns you to the edit line.

Ins	(insert blank lines)

Specify a number of blank lines to insert following the current line. It is suggested that a moderate number of lines be inserted. If you, for instance, insert 23,791 lines then the computer will actually try to create that many lines. It is not necessary to use this function to add lines at the end of a file. The computer will add lines as needed on the end of the file.

Del	(delete lines)

Specify a number of lines, including the current edit line and they will be deleted. Lines are deleted regardless of if they are blank or contain data. TED will add blank lines if the resulting file is less than one screen. LED only requires one line in the file.

Go	(goto a line)

Goto a record (line) number. Enter "0" to move to the beginning of the file. A large number moves to the end of the file. The line will be placed at the top of the screen.

Pgm	(call a program)

The program must be either in RAM or ROM. It will not be found if on mass storage. If the called program ends properly (with END, not an error or pressing **ATTN)** then this program will continue with the cursor on the same line of the file. Be sure, if you call this progrm recursively to edit the same file, not to delete lines and place the previous edit line outside of the file size. When a program is CALLed a new environment is created. Be sure that there is enough memory available. The section on the MENU program has a discussion of calling programs. Not supported by LED.

Bln	(delete blank lines)

Deletes all blank lines and lines containing only spaces from the entire file.

Trn	(truncate long lines)

Truncate all lines to the specified display width. Not as violent as it sounds. What actually happens is that lines longer than will fit on the monitor will be split into two or more lines. Lines are split between words, those with words longer than the display width will not be shortened. This function is helpful if you often use the computer in the field, in which case data may be entered without concern for line length. When you later edit the file with a monitor this function will make it more readable. Since text formatter commands for the REPORT program may be placed anywhere in the line, this function helps make "train of thought" writing easier to read without concern for final printed form. This function will take from a few seconds to several minutes depending on the size of the file and the number of long lines. Not supported by the LED program.

E	(exit)

Exit the program.

V	(video toggle)

Toggle video update on/off. Has virtually no effect if there is no display device. When display update is disabled the monitor is still active, and prompts still appear at appropriate times in the usual location. However, up or down movement will cause the line to be displayed much like using the LED program. Not supported (or needed) LED.

UTIL WorkBook71 Extension Program

The UTIL program provides less-often used functions for WorkBook files. It is an extension of the WorkBook Editor (WB) and may be CALLed from that program, the MENU program,or CALLed or RUN directly from the keyboard. The UTIL program adds functions such as editing label files and inserting Columns in the WorkBook file.

When the program is running the first prompt will ask you for the name of any valid WorkBook file. Enter the name of a file in RAM or on mass storage. You may specify a device name (such as ":TAPE) with the file name.

The program acts as a command level, pressing the highlighted (uppercase) keys enables the specific functions. When the function is completed the main menu will reappear. Note that with the exception of editing Col/Row labels, the ATTN key will suspend the program. The second command prompt may be displayed by pressing any non-command key (such as the un-shifted white keys), the beginning of the prompt displays the name of the active label file, if any.

Add (add col or row)

This function is used to add a new Col or Row at the end (extreme right or bottom) of the file. The file size is increased by that number of cells. Since mass-storage based files must, of necessity, be of a fixed length, you may only increase the size of a RAM-based file. The file size will not be increased if there is insufficient memory to add the complete Col or Row. Note the difference between the Add and Ins commands.

Ins (insert col or row)

Insert a Col or Row at the specified location. Data from that location through to the last Col or Row will be shifted. Any data in the last Col or Row will be lost. Cell refrences are not adjusted.

S&r (search & replace)

Search for a substring in all cells in the file and replace it with a specified string. Note that the replacement will not be made if the resulting string is longer than the maximum cell width.

N (new file)

Change to a new (different) WorkBook file.

E	(exit)

Exit the program.

Lbl	(edit a label file)

Edit or create a Label file for use with WorkBook files. It is not necessary to edit the file associated with the currently active WorkBook file. You may enter more or fewer labels than Cols or Rows in the file (or files) the Label file is to serve.

The first prompt asks you for the name of a label file. Enter the name for a current label file or a valid new file name. Remember, it is not necessary to edit the label file associated with the current WorkBook file.

Col/Row/Menu

The three options are to edit Col labels, Row labels or to return to the main menu. Either Col or Row labels are edited the in the same manner. The prompt will show "col#nn" or "row#nn" followed by a flashing cursor and the current label for that Col or Row, if there is one. The up and down arrow keys move you through the labels. The **ENDLINE** and **RUN** keys will move down to the next label. The **ATTN** key will return you to the label sub-menu. When you press **ATTN** to terminate editing the current Col/Row it will take a few seconds for the program to transform the data to the standard format before the menu returns.

LABEL SPECIFICATIONS

Labels may be up to five characters. The first character may not be a number or any other character below ASCII 65 (see your HP-71 owner's manual for details; basically it means no control codes, spaces, numbers or punctuation, but upper- or lower-case letters are allowed). There may not be embeded spaces. Any characters beyond a space will be ignored. There is no protection against writing multiple identical labels, however, only the first one with a given name will be found by any program. The same label names may be used for both Col and Row, for example, 'TOTAL' could appear for both Col and Row without confusion. Since labels are always compared in uppercase it is suggested that you not use 'total' and 'TOTAL' in the same file because they appear the same to the WorkBook programs. It is not necessary to enter a label for every Col or Row, leave the prompt blank and go on to the next entry. When the WorkBook editor displays labels these will be displayed with their default Col or Row numbers.

Using Labels

Many operations require the refrence to a Col or Row or a combination of the two called a coordinate. Coordinates are always entered with first the Col, then Row coordinate separated by a comma without any extra spaces. The default is

the Col or Row number so you may enter either default numbers or labels or a combination of the two.

```
goto: wk,total              goto: 6,total
view: wk,7                  view: 6,7
+[mon,misc]+[tue,misc]
```

Of course, you may enter an abreviated form of the label, enough to insure that it isn't confused with another label. Since the programs look for labels from the first in the list, the first which matches the abreviated label will be found.

Labels are generally easier to remember than Col or Row numbers and are easier to use without a monitor because it is more difficult to become lost in the file. The examples shown in the PRINTWB section of this manual demonstrate the look of a file printed with user labels.

The Label file is generally fairly small and can be used for several different WorkBook71 files. Regardless of what other files are in the computer, you will probably find that your favorite label file will remain in your 71 as often as WBLEX.

Label File Specifications

The Label Text file is a standard HP-71 Text file which contains four lines. The first line lists the total number of Col labels, the second line is the total number of Row labels in the file. The third line is the actual Col labels and the forth (and final) line is the Row labels. A total of six characters are used for each label: a leading space plus as many as five characters for the label itself. If the label is shorter than five characters then it is filled with spaces on the right. That leading space is mandatory; it is the "hook" used by the programs to find labels and tell the end of one label from the beginning of the next.

Since the maximum line length which may be edited with a Text Editor is 96 characters, you may edit any label file with 16 or fewer Col or Row labels directly using a Text Editor as long as you are sure to maintain six characters including the leading space for each label and keep the totals on the first two lines accurate. Any label file with more than 16 Col or Row labels must be edited with this program.

Since the standard record size is 256 bytes, it is best to use 42 or fewer Col or Row labels per file.

PORTER File Transfer Program

Porter is used to "port" (or move) data between WorkBook71 and Text files. The purpose is to offer compatibility between WorkBook71 and files written with other programs and to exchange data with other computers.

Since fields within the file may be delimited by any displayable character, it is unlikely that there will be problems when exchanging data with, for instance, HP-75 or HP-85 spreadsheets and other programs which use Text files. Remember, if the line length is kept under 96 characters, then the Text output file may be edited with the TED program to correct any unusual lines. The limit on Text file line length is 255 characters, if any source or output lines exceed this length then the program will halt when it comes upon them.

If you need to exchange data with HP-75 VisiCalc (registered trademark of VisiCorp) files, use the SWAP program in "File Management Applications for the HP-75", also by this author.

A disadvantage of using commas to delimit fields is that formulas and text often contain them and each will be interpreted as a field separator and make for a very unreliable transfer. It is suggested that the accent grave (`) be used when practical since this character is rarely used for other purposes.

The PRINTWB program may be used if you wish to copy a WorkBook71 file to Text formatted as it is printed. That is the method easiest to use to incorporate a WorkBook71 file into a report which is also to contain other descriptive text.

If the source file is Text then any blank lines will be ignored. Blank lines will be translated to blank WB cells if fields are ignored, so that purposfully-entered blanks will be retained.

Getting Started

The program may be CALLed or RUN from the keyboard or called from the another program. It is aproximately 2K and uses another 700 bytes while it runs. The source file may be either in RAM or on Disc, you have the option of specifying a device name. Remember, to minimize medium wear, it is a good idea to not have both source and destination (also called "output") files on disc; also, Text destination files must be in RAM. The output file must be a new file name if it is a WorkBook71 file; if it is a Text file then the new data will be appended to the end.

There are several options which have a great influence on the format of the output file. Since the source file is not altered in any way, it might be helpful to experiment with how these various options affect the output file. If you have a Text Editor program then an accompanying LEX file extends your 71 with the capability to LIST and PLIST Text files, this may be done to quickly view the file.

PORTER Commands

The program begins with up to seven prompts (depending on the methods being used). Prompts for a single character input (such as Y/N) must not be followed by **ENDLINE**.

After the final prompt, when the computer does the actual conversion, the current time is displayed. The now-current time is appended to the display when the program ends. Use the difference between the two times to help judge how long the program will take to run. The program takes from a few seconds to several minutes depending on the size of the source file and whether or not the files are RAM based.

```
source file:
```

Enter the name of a Text or WorkBook71 file in RAM, :PORT or on :MASSMEM. The file will be accessed as a virtual file if you specify a mass storage device; it will not be loaded to RAM. If the file is not found or not of the correct type then the prompt will return. The only acceptable file types are Text and WorkBook71 type Data files. Data files formatted using other systems may not be used.

```
output file:
```

Specify a new WorkBook71 file or Text file. Remember, the output file can not be the same file type as the source file (if the source is WB then the output must be Text, and vice versa). If there is already data in the Text output file then this data will be appended to it. If it is a WorkBook71 file then a prompt will follow these next prompts requesting the size and configuration of the new WorkBook71 file.

```
Field delimited Y/N
```

Your answer to this prompt determines what further prompts will be required. This question asks: if source file is WB then do you wish to place all of one col (or row) on the same line, with the data from each cell separated by a "delimiter" character. If the source is Text then **Y** means you wish to place the delimited sections of the file in separate cells. Pressing **N** signifys that you wish all data to be transferred sequentially to separate lines (or cells) without interpreting separate fields. This second method is often used when transferring non-spreadsheet oriented Text between formats. In this case data will be copied beginning in the upper left corner and progressing through the end of the first col to the beginning of the second col, and so forth. Be sure the resulting text file lines will be shorter than 255 characters.

If you have indicated **N** then the following prompts will not be displayed.

```
field separator:`
```

At this prompt specify the character that you want the program to place between fields when transferring to WB files or to interpret as field delimiters when transferring to Text. Specify any character you wish; while only a single character is usually used, you may specify up to 32 characters. The default character is the accent grave (`), this is done because the character is not assigned to a key and this method makes it easier to enter the character. Most programs for other computers use a comma (,) to separate fields. You may wish to change the prompt to this character. The REPORT program, furnished with this package, uses the comma to designate separate fields in Distribution List. Many programs for the HP-75 use the accent grave. Moving data between the HP-71 and HP-85 may be easier using commas.

```
using:Cols/Rows
```

This specifies the direction within the WorkBook file in which data will be written or read. If you specify **C** then each line in the Text file will be transferred to separate cells within a Col (vertically). Specifying **R** will cause data to be transferred between cells in Rows and lines in the Text file.

```
delete formulas Y/N
```

This prompt will only be displayed if the data is being transferred to a Text file. Pressing **Y** will cause only the results of the formulas (as displayed on a monitor or printed) to be transferred. This also causes trailing decimal places (found when using FIX 0) in numerical or formula cells to be deleted. If you press **N** then the entire cell contents will be transferred. Label cells are not affected by this option.

FINDER Program

Searches WorkBook DATA files for user-determined data and provides a TEXT file with the results of the search.

DATA FORMAT

When the WorkBook program is used as a List Manager data is usually stored as a series of columns with each row representing a field. Any number of rows (fields) in the file may be used. An sample data-base file called "DBSAMPLE" is included on the Disc. When fields meet the qualifications you specify, as many rows in the column are copied to a TEXT file. This is called the output file.

The format of the output file is visually the opposite of WorkBook files. That is, columns are reproduced as lines in the TEXT file. Each column in the WorkBook file is copied to the TEXT file as a single line. Rows within that col are separated by the accent grave (`) or other field delimiter as specified. It is suggested that you attempt to keep the total number of characters in each output line to a maximum of 96 so that the lines may be edited with a Text Editor program. This is done by specifying a range of output rows that will definitely fit within the limit.

INSTRUCTIONS

Load the program, It is called "FINDER". It may be run from the WorkBook Editor, a menu program, or run directly from the keyboard. At the first prompt enter a Valid WorkBook file name in RAM or on the first (or only) Disc connected and active. You may optionally include a device specifier (such as ":TAPE").

OUTPUT FILE

The next prompt is

```
output file:
```

Enter a TEXT file name or a new name. Data will be appended to the end of the file if some already exists. The output file is created in main RAM.

OUTPUT ROWS

The prompt is followed by the number of the first and last row in the file.

```
output rows: 1,8
```

You may change the range to any other rows. You must specify two rows, If you wish to use only one row, specify the same label for the whole range, for instance "4,4". The second specified row must either be further down in the file or the same row. You may use either default or user headers, providing that the header file is present in RAM at the

beginning of the program. Valid input can be "1,Zip" or "Name,4" or whatever, providing you spell them correctly.

FIELD SEPARATOR
Specify the character which you wish to be placed between fields in the output file. The default character is the accent grave (`).

```
field separator: `
```

While most programs use commas for field separators, this character is supplied as the default because of the dificulty of entering it from the keyboard, and because often fields may already contain commas. Be sure of the delimiter character needed before running this program.

SEARCH STRINGS
Search for up to 12 different strings. Each may be in a different row. Data may be entered in upper or lower case. Comparisons are made in upper case. Beginning spaces are taken literally, spaces at the end of the string are ignored. The maximum length of the string is 8 characters, including any control codes. The program will look for the position of your string within the cell, it may be in any position within the cell. As an example, if you specify the number "9" as the string and "Zip" as the row, any zip code with a "9" will be found.

SEARCH OPTIONS
To specify the comparison from the beginning of a field, begin the search string with the slash "/". So, if you wish to find those Zip codes beginning with "9", use the combiniation "/9".

To specify comparison from the end of the string, specify, for instance, "123/" and it will make the comparison from the right end of the string. It is not necessary to enter the whole name, a significant portion, enough to insure that it does not look like another, is enough. For instance, to find Los Angeles, you may wish to enter "LOS AN" or perhaps "OS ANGEL". Odds are that there aren't other cities similar to "OS ANGEL". If there are you can eliminate them by specifying Exceptions (more later).

SEARCH STRING SPECIFICATIONS
The search strings do not need to be searched for in the range of cells you specify for the Output file. You may use numbers as well as strings for comparisons. Numbers are compared without the leading "=" or "+". Formulas themselves are ignored. All numbers are compared to the number as displayed in the file.

SEARCH ROWS
The prompt will show the last row used. you may press **ENDLINE** and use that row or enter a new row label.

```
row: 4
```

To search more than one row for the same string it is necessary to use the same string again as a new string, and specifying the next row to be searched.

MANDATORY STRINGS
If you have input more than one search string, the next prompt is:

```
strings mandatory Y/N
```

If all specified strings MUST be present in ALL search fields, press **Y** all other times press **N.**

When you have entered the last search string, press ENDLINE without any input. If you press **ENDLINE** without specifying any search strings, it will be assumed that you wish to search columns for the exception strings.

The next comparisons are made in the cols found by search strings (or all if we didn't specify any strings at all). Pressing **ENDLINE** without any search strings is saying "I want to use all of the data EXCEPT what I am about to tell you".

< OR > COMPARISONS
Less than and greater than comparisons are next. If you don't wish to use them, press **ENDLINE** at the prompt without any input, and the program will move on.

Less than and greater than comparisons compare strings to each other. These comparisons will only be made in cols which have met the search string qualifications above. You may compare numbers, but remember they are strings. "-1" is smaller than "10" because we are comparing ASCII character codes. Numbers may only be compared accuraely if they have the same number of characters. "<STRING" means if the cell contents are smaller than the "STRING". " >STRING" is the opposite. Strings are compared with the same number of characters. If your search string has five characters and the cell has eight, only the first five will be compared. If the cell has two characters and your string has five, only the first two characters are compared. You may use up to eight characters, but the first character must be either "<" or ">" or the comparison will not be made. Up to twelve different comparisons may be specified.

The second prompt for each comparison is:

```
row:
```

Enter any valid row label.

EXCEPTION STRINGS
Using Exception strings is the same as search strings except, of course, if the program finds the string, it will not copy

the specified column to the output file.

You may use up to 12 exception strings and they may be very specific to eliminate small groups from the output file. For example, you may wish to list everyone with "9" in the beginning of "Zip" EXCEPT if they live in the San Francisco area. You would use "/9" as the search string and "/94" as the exception string.

NOTES

- If no search strings are specified it will be assumed that you want to only look for exceptions.
- If no search, < or >, or exception strings are specified, then all fields in the specified range will be copied.
- The program will end if the computer runs out of memory.
- The source file is not changed in any way.
- The examples above include quotation marks for clarity. Do not enter the quote with your comparison strings.
- Make sure you have enough battery power.
- Execution time will vary considerably. For a single search of a small file, expect results in a few seconds. If you are comparing a large variety in a very large file you should plan you lunch break around it.

- This program may create TEXT lines as long as 255 characters. The program will not operate correctly if the total number of characters in a line exceeds 255.
- Each comparison you specify will be made on the entire file before moving on to the next comparison. If the file is cassette based there may be considerable tape wear (in addition to considerable delay).
- Files may be transferred back into the WorkBook format with the PORTER program. That program may also be used to reverse the col/row orientation.

SORTWB Sorting Program

Often we wish to view data in a sequence other than as it was entered. Most often we will want to alphabetize by a single data item such as last name or Zip code. Numeric values are stored in WorkBook71 files in much the same form as Text so they may be sorted in the same manner. This program has an unusual option in that you may specify which word in a field (cell) to use. This is helpful if you have a single field with, for instance, both first and last name; you may specify the second word and the program will sort by the last name. To eliminate any problems with data which may have been entered with leading or trailing spaces, any extra spaces are ignored and have virtually no effect on the speed of the program. Unlike most sorting programs, empty fields will be placed at the end of the output, not the beginning.

As with many other programs in WorkBook71, there are a number of different ways of handling the data and these various methods have a considerable impact on the outcome of the program. Please read this section and experiment with the options available before using the program with inportant data. Execution time can take from as little as 15 seconds to several hours; be sure you have a reliable power source for all devices connected to your 71.

Sorting Methods

Two different methods of sorting are available with this program. A Bubble sort allows data to "float to the top" while the Quick sort works to the divide and conquer as fast as possible. Quick sort is much faster when working with large quantities of data. The primary advantage to the Bubble sort is that items within a given range, say "Smith" remain in near sequence they were entered while Quick sort will organize the Smiths in a seemly random order, though all grouped together. If the quantity of items is small, say 50 or fewer, or maintaining the sequence of items within a series is important then the Bubble sort is preferred. When dealing with a large quantity of items then Quick sort will be the obvious choice. The following table gives an approximate speed comparison for the two methods. Note that the time is for the actual sort only and does not include the time to read files and output results.

# Items	10	50	100	500
Bubble	3	50	200	5000
Quick	5	30	60	400

As you can see, Bubble sort would take aproximately 5000 seconds or over an hour and twenty minutes to sort 500 items while Quick sort takes about six and a half minutes.

How Data is Interpreted

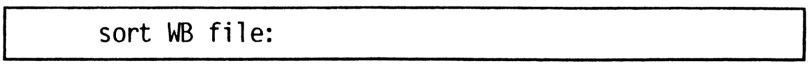
All sorts are done in ASCII alphabetical order, regardless of the data type. If the field contains a number (the field begins with a plus or equals sign), then only the number itself will be used. Text fields will use only a single word within the field; the default is the first word. Comparisons are made without respect for upper or lower case. In ASCII order sorts, numbers preceed the letters A-Z, but follow most punctuation. ASCII characters and their values are listed beginning on page 322 of the "HP-71 Refrence Manual". All numbers should contain the same number of digits for proper sorting since shorter cells will be placed before longer cells. Blank (empty) fields will be placed at the end of the output.

How the Program Works

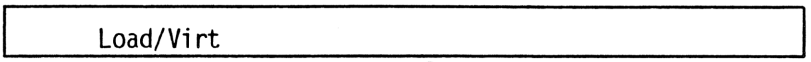
One problem with sorting programs is the constant movement of data. This can cause a considerable speed loss as well as extra wear on the Disc media (if a mass storage based file is being used). This program begins by reading the data from the specified range of cells into an array then sorts just a single word and data pointer for each field. Once the data in this array is sorted, the data pointers are read and the resulting output is prepared.

As an example, let's say you have a WorkBook file set up as a data base; each col represents an entry, and cells within that col represent fields. The file has six rows (or fields) and 100 cols (or entries). The first row label is "Name" and cells within that row contain first and last names, separated by a space (or multiple spaces). When you run the program you specify "Name" as the row and the range to include the entire file (cols 1-100). When the prompt asks, you specify the second word (for the last name) in the cell. If the file is RAM-based it will take about 18 seconds to load the cells into the array. The program will display the current time and a prompt to tell you what operation is being performed. Once the array is filled the prompt will display the now current time (about 18 seconds later than the first time), then a prompt to tell you that it is now sorting. Using the Quick-sort, the file will be sorted in just about one minute. At this point the prompt will again change to tell you that the program is now writing the sorted output as you have specified.

The SORTWB Prompts



Enter the name of a WorkBook71 Data file in RAM or on mass storage. You may include a device specifier with the name. If the file was not found the following prompt will be displayed.



Pressing **L** will cause the computer to try to load it to RAM. **V** will cause the 71 to search for the file on the first (or only) mass storage device.

```
sort on row: 1
```

Since a WorkBook file used as a data base is usually set up with one full column for each entry, fields are represented as rows. At this prompt enter the row, specifying user (if an apropriate label file is in RAM) or default row label.

```
range of cols: 1,8
```

The sorting can be restricted to a limited range of the file. The default is from the first Col to the last Col in the file. Press **ENDLINE** to use the entire file or change one or both parameters to use only a portion of the file.

```
max len: 5
```

The sort is always on a single word. You may set the precision of the word at this prompt. The default is five characters; you may change this to any reasonable value. For example, sorting by state will usually only require two characters, while sorting by Zip code works best with five and using last names can be as long as twelve characters or more. It is not necessary to set this value to the full length of the word; you could sort by just the first few characters of the last name. Memory consumption during the sort increases with the maximum length, though this memory is reclaimed before forming the output.

```
word#: 1
```

If a single field contains several words you may choose which word within that field to sort on. A word is considered as any group of characters set off by spaces or at either end of the field. The following are examples of fields which you may want to sort by either the first or second word at different times. Note the City, State field contains a comma, but also a space following the comma to separate the items.

First Last	number street
City, State	Item code#

If the field contains fewer words than the number you specify then that field is considered to be empty.

The Output File

The final prompts ask how you would like the output formatted. You may send the data to a new WorkBook71 Data file or a Text File. If the output file does not exist you will be asked

whether you would like it to be Text or WorkBook Data, and the file size. The file may be in RAM or on mass storage. If the output file is Text and on mass storage, be sure that it is large enough to contain the data created. If the output file is not large enough to contain the entire source then as much as will fit will be returned.

If the output file is Text then you have the option of appending the data to the file or overwriting existing data. Text files in RAM can be of any size since they will expand to hold the data.

MENU Program

This program is an extension to the WorkBook71 Spreadsheet (WB) with three main purposes.

1-To simplify running programs by virtue of a friendly user interface.

2-Conserve memory by loading applications from Mass Storage only when needed.

3-Insure that the LEX file WBLEX is in the computer before Calling programs.

MENU operates as a high level shell something like a simplified PAM (Applications Manager) used on some larger Hewlett-Packard computers. As furnished, MENU is set up to CALL the various modules of this system. Through simple programming it can be extended to list any program you wish.

Using MENU

The program is compact and may be kept in :MAIN or :PORT RAM. The program may be either RUN or CALLed as desired. When the program starts you will be presented with a Command-level type prompt followed by a file name. If a display device is available a complete list of the available files will be displayed and the prompt will be near the bottom of the display. A large number of programs in the MENU will cause the display to scroll, but will not affect program performance. The first thing the program does is search for WBLEX. If it is not found then MENU will try to load it from Disc. If, for any reason, WBLEX cannot be found, the program will end.

MENU is Command-driven. All options require a single keystroke to innitiate. The **ON** key can be used to temporarily suspend the program however, **E** should be used to terminate the program.

MENU Commands

C (cat all)

Do a CAT ALL of :MAIN and :PORT RAM. You may press **f-LINE** to move to a different port, but do not press **f-EDIT** as it will cause the program to halt with an error.

M (cat of mass storage)

Does a CAT of the first or only mass storage device.

f-CONT (continue)

Rebuild the display and show command prompt.

UP ARROW

Move up one name in the menu.

DOWN ARROW

Move down one name in the menu.

ENDLINE

CALL the program currently in the display. If the program is not in RAM then MENU will try to load it from Disc, then purge it from RAM when control returns to MENU. You should always be conscious of available memory before using any program because memory requirements often increase as programs run.

P (call a program)

CALL a program not on the menu. Will ask for the name then CALL the program in the same manner as **ENDLINE.**

Returning to MENU

Any programs called from MENU should end with an END or END SUB statement. If the program is suspended by a PAUSE statement or an error then MENU will not continue. Terminate the program and return to MENU by entering "END @ CONT" from the keyboard and pressing **ENDLINE.** If the menu does not reappear then you will have to enter END ALL. You should also edit the problematic program to determine what went wrong.

Extending the MENU

Besides being able to call programs using the **P** key you may add your own programs to the MENU. Line 20 in the program assigns the number of programs on the MENU to the variable N. As furnished the value is 9 for the 9 main modules (excluding this program) of WorkBook71. The actual menu begins on line 21 and can extend to as many as 98 programs (in which case it would be faster to enter the program name than to find it in the MENU). Let's look at the beginning of the MENU program listing.

```
10 CALL MENU @ SUB MENU
20 N=9 ! number of items in menu
21 DATA WB WorkBook71
22 DATA TED text editor
23 DATA LED line editor
24 DATA REPORT text formatter
25 DATA PORTER file converter
26 DATA FINDER file searcher
27 DATA SORTWB sort wb files
28 DATA PRINTWB print wb files
29 DATA UTIL wb71 extension
30 ! then next item is added here
40 ! program begins here...
```

Let's say you want to add the program "TVM" from the HP Finance Pac to the MENU. Line 30 (the next available line) would be entered beginning with a DATA statement then the program file name followed by a short description. Attempt to keep the program name and description under 22 characters for readibility.

```
30 DATA TVM time val of money
```

In order for the program to recognize the new entry change line 20 to reflect the new addition to the MENU.

```
20 N=10
```

The variable N on line 20 should always account for the actual number of MENU programs. The exact line number is not important as long as the first entry begins on line 21 and line numbers which are part of the MENU program may not be used.

CALLing Programs with Parameter-Passing

Some programs require parameters to be furnished when you first CALL them. A simple method for providing these parameters is by using an intermediate program, the purpose of which is only to supply the data then CALL the program. As an example, several Hewlett-Packard ROMs have a Text-Editing program Called EDTEXT (your ROM may have a different name for it). The program requires at least one parameter: the name of the file you wish to edit. A second optional parameter is how you would like to enter the program. We'll asume that we want to edit the file, which HP has decided needs the code letter "T".

This intermediate program should have a file name different from the name of the CALLed program. In the example we will Call it "ED" because that is a common name given to Text Editors.

```
10 CALL ED @ SUB ED ! a separate program environment
20 INPUT 'text file:';Q$ ! get file name
30 CALL EDTEXT(Q$,'T') ! call the program
```

As you can see, all it does is ask for the file name then in turn supplies the name to EDTEXT as a parameter. A simpler program can be used if the parameters are always the same. In this case the program could be a single line which supplies the data the same as if you were Calling the program from the keyboard. Remember that a single program file may contain several SUB programs if needed.

WBLEX Utility LEX File

This custom LEX file was written for use with WorkBook71. It is a group of keywords which expand HP-71 BASIC with new forms of INPUT, data formatting and utilities to speed formula recalculation string extraction. You may find some of these keywords helpful in programs written for your own use. You may not distribute programs using this LEX file without express written permission of the author. The LEX ID is 61 (hex) with keywords beginning at 3D. These descriptions are informational only, no guarantee of operation is made.

**CELVAL$** Q$=CELVAL$(Q$)

Cell value. This is the most specialized function in WBLEX and is probably of limited usefulness elsewhere. If the first character in the string is a plus or equals (+=) then the next word, up to the next space in the string will be returned (without the +=). Zero length strings and strings not beginning with + or = will return the string "0". It is used by the WorkBook Editor during recalculation.

CLFLS

Close Files. A statement requiring no parameters. Does an ASSIGN #n TO * on all currently assigned files. This is helpful because the HP-71 does not automatically close files when exiting a SUB program. More memory-efficient than individually closing even one file.

**CUE$** Q$=CUE$(S$,C)

A specialized form of INPUT used as a function instead of a statement. S$ represents the default input string, C is the location within the string where you would like the cursor to begin at. Valid cursor position is 1-95. If the string is shorter than the cursor position specified then the computer will place the cursor at the end of the string.

The command stack is not enabled using the up/down arrow keys; instead they are terminator keys. The function returns a single character string representing the actual ASCII code of the terminator key, not the keymapped value used by the mainframe KEY$ function. This function is compatible with WTKEY$.

Terminator	CHR$	Terminator	CHR$
ENDLINE	13	g-CURSOR UP	20
ATTN	14	g-CURSOR DN	21
RUN	15	g-ATTN	22
f-CONT	16	f-CALC	23
f-SST	17	OFF	24
CURSOR UP	18	g-ENDLINE	25
CURSOR DN	19		

When CUE$ terminates, it returns the terminator key, the edited string may be recalled with the DISP$ function. The location of the cursor at the end of input is returned in the RES register and may be recalled using C=RES if done before the next mathematical function.

**ECP$** E$=ECP$&CHR$(C)&CHR$(R)

Escape Percent. Requires no parameters. Returns the two character string CHR$(27)&"%". This is the beginning of the escape sequence used to move the cursor to a specific col & row on the display.

**FMTCEL$** X$=FMTCEL$('string',width)

Source	Returns
"=123, ????",12	" 123"
"ABCD",10	"ABCD "

Format Cell. Used to format strings for display or printing. Specify a string and the number of characters in the string to return. The string will be filled or truncated to the specified width.
The string will be left-justified with as much as will fit within the width specified. If the width specified is greater than the length of the string then the string will be filled with spaces on the right. If the string begins with a plus or equals (+=) then that character will be deleted, the string will be truncated to the next space, and then right-justified. When the string is right justified, if the last character is a period it will be deleted; this is done because FIX0 returns a trailing period. If the string had been right justified but was too long to fit within the specified width then a '>' and as much of the string as will fit will be returned.

To sum what happens, the string will be left-justified to the width specified. If the first character is + or = then it will be assumed that the string represents a number and possibly some following unwanted data (usually a formula) so it will be trimmed accordingly and right-justified.

**HI$** S$=HI$(S$)

Sets the high bit on all characters in a string. Useful for enabling the alternate character set on the computer's LCD and for displaying inverse video on a monitor.

NVAL X=NVAL('string')

Source	NVAL	VAL
"123"	123	123
"-2"	0	-2
"2*2"	2	4
"ABC"	0	(var contents)
"" (null)	0	(error)

Number Value. A simplified version of the VAL function. First eliminates leading spaces then reads the string literally and terminates when the first non-integer character is found. Negative numbers, non-numerical, null or strings with greater than twelve digits return zero. Since it will not return negative values, it qualifies data and simplifies boolean arguments. Will not cause an error with bad data.

**NXTWORD$** R$=NXTWORD$(S$)

Source	Returns
" ABCD"	"ABCD"
" ABCD EFG"	"ABCD"
"ABCD"	"ABCD"
" "	"" (null)

Next word. Returns the next series of non-space characters in a string. Ignores leading spaces and will return a null string if the source string contains only spaces.

PAK$	R$=PAK$(S)

Pack number. Supply an integer from 0-65535 and it returns a two-character "packed" representation of the number. This function is the opposite of UNPAK.

REVRS$	R$=REVRS$(S$)

Source	Returns
"ABCDEF"	"FEDCBA"

Reverse string. Reverses the order of characters in the string.

TVIS	T=TVIS

TV is. Requires no parameters. Returns the location of the currently active display device on the HP-IL. If there is no active display device, OFFIO, CONTROL OFF, or there is no HP-IL module it returns zero. This function has been tested with standard HP and RS-232 display devices. This function is not useful for display devices assigned with extended addressing.

UNPAK	R=UNPAK(R$)

Unpack. Reads two characters from the string and returns a numerical value from 0-65535. The opposite of PAK$.

WTKEY$	K$=WTKEY$

Wait Key. Places the HP-71 in a low power state and waits for the next keystroke. Returns the uppercase value of the key regardless of the LC setting unless the user presses the [g] key first. Returns the ASCII value of any key, not the keycode. For instance, ENDLINE returns CHR$(13), not "#38" which the HP function KEYWAIT$ returns. Note that the LC setting inverts the case of the letters A-Z when using an external keyboard.

WTRIM$	R$=WTRIM$(S1$)	R$=WTRIM$(S1$,S2$)

Source1	Source2	Returns
" ABCD"	" A"	"BCD"
" ABCD"	" A????"	"BCD"
"ABCD"	" ABCD"	"" (null)
" ABCD"	(none)	"ABCD"

Word Trim. Accepts one or two parameters. If only one string is specified then it trims leading spaces. When two strings are specified then it first trims leading spaces on both strings then compares the two strings. The first string will be trimmed by as many characters which match (in the same relative position) in the second string.

The PRINTWB Program

PRINTWB prints WorkBook Files in a grid of columns and rows, much as they are displayed on a monitor, or they may be listed sequentially with formulas.

Column and row labels cannot be printed "inverse" with most printers so they are delimited with vertical bars "|". In addition to this difference you can supress labels, print sections of files and use a different Col width than you might use with a monitor. Listing a file in unformatted form is helpful to review formulas or to list data-base files. When printed unformatted a blank line will be printed after each column.

WorkBook71 files may also be printed to a Text file. This is useful for embelishing a report before actual printing or merging with files printed by the Text-Formatter program.

If you need a WorkBook71 file transferred to a Text file for the purpose of exchanging data with other computers use the PORTER program.

When you use the WorkBook Editor (WB) with a monitor you will notice the display builds by columns, from left to right. This is done to try to keep the process as fast as possible. As you will recall from the discussions of formulas and recalculation, this is because the file is being read sequentially. Now we come to this program. When formatting the file, it is necessary to print one entire Row before printing the next; most printers can't back-up. While it is permissable to print formatted Virtual files with this program, they will print much slower than RAM-based files because of this method of reading data from the file. When using the program to list a file and its formulas, it is read by cols and there is no speed (or media wear) penalty.

The Printer

The program does not assign the printer. Whichever device is assigned and active when the program begins will be used. If there is no printer then the output will be sent to the LCD and, if there is one, an active display device. The ThinkJet and other Hewlett Packard printers usually are automatically assigned as the active printer by the HP-71. Printers using RS-232 or other interfaces must be assigned, Please read the HP-IL Module owners manual for instructions. Be sure, before running this program, that you know the number of characters your printer can print on one line.

Run the Program

PRINTWB may either be RUN or CALLed; it requires no parameters. The WBLEX LEX file is used extensively this program. Be sure you have enough memory for the program, variables consumed, and room for any output file created. In addition to the size of the program and variables used, this program uses a scratch buffer in RAM which is twice the size of the specified col width. Once the program begins you will be lead through a series of prompts.

PRINTWB Prompts

The program takes you through as many as eleven prompts about how you would like to print the WorkBook File. The program can print the file in many different forms. Your response determines how the final printed format will look. Many prompts begins with a default value, such as current PWIDTH and range of cells to print. You may may press **ENDLINE** and use the values offered, or change them. Others require a single keystroke such as after "Y/N"; in that case it is not necessary to press **ENDLINE**. If, at any time when entering data at these prompts, you become lost, you may suspend the program by pressing **ATTN** then restart it with the **RUN** key. Now, let's work through the prompts:

```
print WB file:
```

Enter the name of a WorkBook file in RAM, :PORT or on Disc. If you specify a device name the file will be read directly without loading.

```
Load/Virt
```

If you had not specified a device name and the file was not found in RAM you have the option of loading it or using it from Disc.

```
start col,row:1,1
```

This prompt is asking for the Col,Row coordinate of the the first cell you would like to print. The default is the the first cell: 1,1 in the upper left corner of the file. You may enter coordinates using default or user labels (if the label file is in RAM).

```
last col,row:8,8
```

The default is the last cell in the last Col; that is, in the lower right corner. If you had pressed **ENDLINE** at this and the previous prompt without changing the values then the entire file would be printed.

You may specify any range of cells, representing a square or rectangular portion of the file.

```
print labels Y/N
```

You have the option of printing the file with user (if a label file is present), default (if no label file), or without any labels at all. This third option is specified at this prompt. Pressing **N** will cause the file to be printed without any labels. This will cause the cells to be printed in order, but without any printed indication of Col/Row specifiers.

```
format: Y/N
```

Press **Y** to have the file formatted in a grid of cols and rows. Press **N** to list entire cell contents, including formulas, by cols. If you specified unformatted listing then the following two prompts will not be shown.

```
col width:8
```

The default is the width specified with the WorkBook71 Editor Command level "Wid" operation. The width you specify will not change the width used by WB, it is only used during this printing session. This is helpful if you normally keep a fairly narrow Col width to display a lot of data on a monitor, but prefer to have a little more room with printed reports.

```
printer width:96
```

Set the maximum number of characters that can be printed on one line. The default is the current PWIDTH setting. The PWIDTH will return to the original setting when this printing session ends. This also sets the line length when printing to a Text file.

The PWIDTH setting is very important when printing WorkBook Files which have too many Cols or too great a Col width to fit on a single page. The program will print the Col and Row labels and as many Cols as will fit on one line. Any Cols which did not fit will be printed, with Col and Row labels, following that section. This is especially important to remember when using a sheet-fed printer. Be sure that the col width specified plus the row label width is less than will fit on a single printed line.

As an example, assume we have a 24 column printer. We have a WorkBook file with 5 Cols and 6 Rows, and Col width is set at 7. Row labels will take 8 characters (5 chars for the label plus a space, a vertical bar (|) and another space), that leaves 16 characters. Only two 7 character Cols will fit within 16 characters so the file will be printed in 3 sections. The same logic holds true for a 132-column printer, but on a grander scale.

```
printer set-up:
```

You may specify anything, up to 96 characters, to be printed on the line immediately preceeding the report. This is useful to add a note such as "summary of January Sales" or for sending non-printing control codes to the printer to alter printing modes before printing the file. Escape sequences may be included by using the exponentiation (^) character wherever a CHR$(27) is required; the program will then replace each with a CHR$(27) when it prints the set-up string. It is not necessary to include any set-up string, in that case press **ENDLINE** without any input.

```
output to:Printer/File
```

Press **P** to send the file to the printer. If you press **F** then the next prompt is:

```
output file:
```

Enter the name of a Text file in RAM or a new Text file name. If the file already exists then this WorkBook data will be appended to the end. The Text file will receive the data exactly as it would be printed. This is helpful if you wish to add a formatted WorkBook file to a printout made with the text formatter.

If you wish to use a mass-storage-based Text file then it must already exist and be created to a size larger than this printed output will occupy.

Notes

At this point, printing will start and will proceed automatically until the file is printed as specified. There is purposfully no error trapping as the program runs. If the program halts during the run it is likely that it is because the output file was mass storage based and was full, there was insufficient RAM memory, a device failed or a loss of continuity of the HP-IL (a plug was pulled). Be sure you understand the cause of any error conditions before running the program again.

User Labels

	Mon	Tue	Wed	Thu	Fri	WkTot
City	PALO ALTO	SAN JOSE	MONTEREY	PALO ALTO	HOME	
Hotel	50.88	40.28	78.81	40.28	0.00	210.25
Trans	2.05	0.00	43.60	4.10	137.30	187.05
Auto	12.00	0.00	14.50	10.40		36.90
Phone	5.10	14.55	3.60		12.40	35.65
Food	16.30	3.27	42.60	16.20	7.00	85.37
Misc	27.50	4.60	43.00			75.10
Total	113.83	62.70	226.11	70.98	156.70	630.32

Default Labels

	1	2	3	4	5
1	Bissenette Ma	BYTE	Calumet	Cartoon Colou	Chicago Broth
2	789 Side St	70 Main St	890 Supreme	D9024 Lindblad	456 Main
3	Philadelphia	Peterborough	Bensenville	Culver City	San Diego
4	PA	NH	IL	CA	CA
5	34567	03458	60106	90230	92003
6	Produce	Publisher	Purchasing	Purchasing	Delivery
7	Claude Bissen	Gene W. Simps	Sir	Linus Van Pel	Luigi

WITHOUT LABELS

PALO ALTO	SAN JOSE	MONTEREY	PALO ALTO	HOME	
50.88	40.28	78.81	40.28	0.00	210.25
2.05		43.60	4.10	137.30	187.05
12.00		14.50	10.40		36.90
5.10	14.55	3.60		12.40	35.65
16.30	3.27	42.60	16.20	7.00	85.37
27.50	4.60	43.00			75.10
113.83	62.70	226.11	70.98	156.70	630.32

Bissenette Market
789 Side St
Philadelphia
PA
34567
Produce
Claude Bissenette

BYTE
70 Main St
Peterborough
NH
03458
Publisher
Gene W. Simpson

Calumet
890 Supreme Dr
Bensenville
IL
60106
Purchasing
Sir

Cartoon Colour Co
9024 Lindblade St
Culver City
CA
90230
Purchasing
Linus Van Pelt

NOT FORMATTED

WITHOUT LABELS

Bissenette Market
789 Side St
Philadelphia
PA
34567
Produce
Claude Bissenette

BYTE
70 Main St
Peterborough
NH
03458
Publisher
Gene W. Simpson

Calumet
890 Supreme Dr
Bensenville
IL
60106
Purchasing
Sir

Cartoon Colour Co
9024 Lindblade St
Culver City
CA
90230
Purchasing
Linus Van Pelt

WITH USER LABELS

```
Mon  ,City  PALO ALTO / HYATT
Mon  ,Hotel =50.88
Mon  ,Trans =2.05
Mon  ,Auto  =12.00
Mon  ,Phone +5.10 @.25+3.8+.8+.25
Mon  ,Food  =16.30
Mon  ,Misc  +27.50 @27.5!LUGGAGE REPAIR
Mon  ,Total +113.83 @[1,1]+[1,2]+[1,3]+[1,4]+[1,5]+[1,6]

Tue  ,City  SAN JOSE
Tue  ,Hotel =40.28
Tue  ,Trans
Tue  ,Auto
Tue  ,Phone +14.55 @3.3+.25+2.6+8.4
Tue  ,Food  =3.27
Tue  ,Misc  +4.60 @4.6!BATTERIES
Tue  ,Total +62.70 @[2,1]+[2,2]+[2,3]+[2,4]+[2,5]+[2,6]+

Wed  ,City  MONTEREY
Wed  ,Hotel =78.81
Wed  ,Trans =43.60
Wed  ,Auto  =14.50
Wed  ,Phone +3.60 @3.6
Wed  ,Food  =42.60
Wed  ,Misc  +43.00 @43.!SCUBA RENTAL
Wed  ,Total +226.11 @[3,1]+[3,2]+[3,3]+[3,4]+[3,5]+[3,6]
```

REPORT Printing Program

This program prints an attractive formatted report from your Text and WorkBook71 files. The effective maximum report size is the size of mass storage, the practical limit for a document is the size of one mass storage medium. When using the 82161A Cassette drive this is about forty pages, the HP9114A Disc Drive could conceivably provide two hundred pages. A fair assumption is that a printed page represents about 2500 Bytes.

Practical uses for this program include printing personal letters, memos, form letters, simple reports, and extensive reports (merging files and taking advantage of your printer's functions). This manual was printed with this program as was the book "The BASIC HP-71", also by this author.

REPORT reads the data in your file, printing the data using commands you have placed within the text. Text may be entered free-form in the Text file and the program will organize it into neat paragraphs during printing. Commands tell the program to center a line, indent for a paragraph or merge another file. These commands are actually the backbone of a simple Text-Formatting language. The original Text file is not altered by this program.

This program will read Text files written by TED (the Text Editor program in this package) or by other programs such as EDTEXT (available in several HP ROMs). Remember, the main file must be Text, so, if you use WorkBook DATA files for the main file, it is necessary to transfer the file to Text (with the PORTER program) before using it as the main file whith this program. This is only necessary if you are going to use the WorkBook Files as Text; The program can read cells directly from WorkBook files without any data format conversion.

A discussion of usage of each command is listed in the Command Summary, later in this section. Please read that section for an overview of other operations not covered in this introduction.

Even if you have the HP Text-Formatter ROM and rarely need to use WorkBook data in your printed reports, you should consider using this program. Besides offering greater printer control, REPORT has the advantage of printing at over 300 words per minute, much faster than the HP program.

A Little More in Depth

Files must be either in RAM, or on the first (or only) mass storage device connected to the computer on the Hewlett-Packard Interface Loop (HP-IL). If you wish to use files from Disc be sure to include the ID of the Disc with the file name. They will be read from Disc (or Cassette) without first loading them to RAM. If the device ID is not specified then the program will search through RAM for the file;then, if it is not found, attempt to load it from Disc. Any files loaded by the program will be purged from RAM when they are no

longer needed. It is possible to have the Main file, a merged file, and a WorkBook file all being read from Disc at the same time. For this reason it is best to have only one file refrenced at a time from Disc/Cassette to avoid continual movement of the medium as sections of one, then another file are read. Be sure you have enough available memory to run this program and the memory it uses for variables. With only this program in RAM, a 17.5K computer will have about 8000 Bytes free for files.

Writing Large Reports

The methods for printing reports larger than the RAM capacity of HP-71 are fairly simple. The following discussion is not necessary if you are writing letters or simple reports; however, please review it for a general overview of the programs capabilities.

Very large reports can be written by keeping them modular or on Disc. This is also helpful if you do not wish to delete other programs and files from your HP-71 to make room for the multiple files, or you have standard sections (like a logo) that you do not wish to keep in the computer. A main file can be used which lists the names of the modules to be merged. The main file will stay in the computer throughout the run, merge files will be located, and read before going on to the next file. Any files that were found in RAM will be left there. CATALL might look like this when you run the program:

```
Name        Type    Length
keys        KEY        256
MYPROG      BASIC     1300
WBLEX       LEX       1024
REPORT      BASIC     5000
MAINFILE    TEXT      3200
MERGE1      TEXT      3000
WRKBK       DATA      1600
```

The file called "MAINFILE" is the name you would enter at the prompt at the beginning of the program. This is file which the program will use to find all files to merge. The file "MERGE1" is read from RAM then the next file called, perhaps, "MERGE2" might be read from Disc printed, and so on. Of course it is not necessary to use these file names. Main and merge files are always Text files. Files merged must be on the first (or only) mass storage device, or in RAM. They will not be found if stored on Magnetic Cards. This program has been tested with the HP82161A Cassette and 9114A Disc Drive. Other mass storage devices supporting the same high level I/O standards should also be usable. If all files are in RAM it is not necessary to have any mass storage connected.

The same general approach is used when recalling data from Worksheets. The program does not require the WorkBook file to be in RAM. Only one Worksheet may be assigned (using the ^WA command) at a time. Whenever data is recalled from

cells it is again interpreted so that you could, for instance, place printer control codes in cells, then refrence those cells when you need the printer code. Cell [1,1] could contain ^ec& ^cck1S to set expanded pitch on a ThinkJet printer. Cells could also contain data such as a greeting or return address. Of course, you can still use a WorkBook file for its intended purposes.

At the extreme, it would be possible to have a main file, a Text file being merged, a WorkBook worksheet all assigned at the same time. Planning and moderation are in order if you are doing an elaborate report.

About your printer

REPORT does not assign nor change the assignment of the printer. Data will be sent to whatever device you (or the computer) have designated as the printer. Be sure that the device you have assigned as a printer can accept the data. If you have an HP printer connected (such as the ThinkJet) then the 71 will automatically assign it as the printer at power-up. Note how the following HP-IL commands will affect output which is destined for a printer.

```
PRINTER IS PRINTER   Send data to printer
PRINTER IS RS232     Send data to Interface
PRINTER IS NULL      Ignore any printer data
PRINTER IS DISPLAY   Send data to monitor
PRINTER IS *         Sent data to LCD display
```

You may test a file by "printing" it to a display device or by designating PRINTER IS NULL to run the program without any printing at all.

It is beyond the scope of this manual to include instructions for every printer. Unfortunately there is little standardization for commands used by printers of different manufacture. The examples shown assume that your printer has relatively few extra features, prints 80 characters per line, 66 lines per page as its default mode and doesn't automatically skip perforations. There are commands you may enter in your Text file if your printer does not follow this standard.

Read your printer manual for codes to change type size, style and number of characters per line and page. There will probably be some sort of cryptic chart listing most of the commands. Often the commands are listed in "hexadecimal" (base 16) format. The function "HTD" will convert the codes to their decimal equivalent and you can find the appropriate key to press in the table on page 322 in the HP-71 Refrence Manual. The easiest way to enter often used (and difficult to remember) printer codes is to include them in a macro file and refrence that file with the user macro command (^U) when you need the code.

Some printers use characters above 127 decimal for graphics. This may be to your advantage if you want to use those characters. For many uses, such as listing programs, it might be preferable to ignore these graphic characters. This can often be accomplished by telling your printer (and interface) to use only 7-bit data. With the HP-71 characters above 127 are displayed as the same character, minus 128. You can assign control codes to keys to simplify entering them.

Printer Graphics Commands

Using Underlined or graphics characters may not respond as expected when they are continued on a second line. The problem is that when the computer indents to the left margin, it does so with spaces. Plan it so that your printer goes out of graphics mode at the end of each line, and re-enters it after the margin has been printed. For this reason, copy mode is preferred when printing graphics. Copy mode ignores the current right margin setting, though follows the left margin. Because of this a graphic can contain a maximum of 255 characters per line if entered in copy mode. Graphics mode on your printer will probably interpret those spaces as the code 32. Usually, when you set a printer to underline, it will do so until instructed not to. This means that the spaces from the left of the page to the left margin will be underlined. An easy way out would be to make sure that all underlined in a series are on the same line.

Syntax Requirements

This program reads Text files and formats reports following the commands you have imbedded within the files. To differentiate the commands from Text to be printed each command begins with the command character (^). Commands may be placed anywhere on a line. The exception to this rule is when you are in center (^CE) or copy (^CO) mode when commands must be placed at the beginning of the line. Of course, if a line is to be copied as it is, then imbedded commands would be ignored. There is no limit to the number of commands which may be on a line. Commands may immediately follow other commands or numeric parameters, but a space is required between data and a command which is to follow it. Several commands have mandatory or optional parameters which may follow on the line. When the parameter is a number (such as page lenth) then no space is needed following that parameter. When the parameter is a formula or file name then a space (or nothing else at all) must follow that parameter on the line to separate it from any following data. A space may always be used to designate the end of a parameter or to separate two parameters for the same command. Commands and parameters may be entered in either upper- or lowercase.

Acceptable:	^PAText	^SK^PA
	^SK2^PA5^DA	^ma8 72^DA
	^ce^da	^UPGM ^PA

```
Unacceptable:    Text^PA          ^SK Text^PA
                 ^CE Text ^FI     ^UPGM^PA
```

Remarks
Any line in a Text file which begins with an exclamation mark (!) will be ignored so that you may embed non-printing information as a note or remark. Exclamation marks which occur anywhere else but as the first character on the line are interpreted as that character and are printed.

Spaces
When printing, extra spaces between words are ignored; a single space follows each word. You may enter extra spaces between data as desired in the source Text file; they have virtually no effect on printing speed. Extra spaces may be helpful at times such as indenting lines to organize thoughts or designate places where data could be added later.

Two spaces are added following periods, exclamation marks, colons and question marks (.!:?) if they come at the end of a word or if they stand alone.

A long line with many words separated by multiple spaces will print faster than several shorter lines containing the same Text.

Fixed Spaces
Many times you will want to be certain that two words will not be split between two lines. Wherever REPORT finds the fixed-space character (~) it will print a space. This is not a character that is normally part of your HP-71 keyboard. This character may be included at any point where you would like an extra space to be printed when it normally would not. As an example of usage consider "Portable~Computer". Any number of fixed-space characters may be used. To assign the key to, for instance, the f-shifted minus key use the following:

```
DEF KEY "f-",CHR$(126);
```

Of course, you can assign the character to the key of your choice. Be sure to end the DEF KEY statement with a semicolon (;) to designate it as a typing aid key assignment.

Getting Started
The program is called "REPORT". It may be CALLed from the WorkBook editor, a menu program, or RUN or CALLed from the keyboard; it requires no parameters. Be sure WBLEX (the custom LEX file furnished with WorkBook71) is in your computer before running the program. REPORT requires aproximately 1100 bytes of memory as it runs. When WorkBook DATA files or distribution lists are used with this, program memory usage increases for such things as pointers and any user Labels.

Line Length Limits

While HP-71 Text Editors limit line length to 96 characters, data entered by other programs or files loaded from other computers may have longer lines. For this reason this program will work with Text file lines as long as 255 characters.

The 255-character limit also applies to the current line being interpreted. Since a user macro (^U) may alter the length of a line, keep this limit in mind when creating macros. As a line is interpreted the results of macros are added to the line and are again interpreted. Since macros allow some recursion, the resulting line could possibly contain more macros which would again add to the line length.

The maximum line length which you can set with the margin (^MA) command is also 255 characters. This potential length is reduced by any non-printing control codes which may be on the line, though does not include the spaces to the left of the left margin. The actual maximum line length is the sum of the right margin minus left margin minus the total number of control codes. In actual use the line is most often aproximately 60 characters. The limit will never be of any consideration except when using a printer in condensed mode with an extremely large number of printer control (or graphics) codes.

Set-up Printer

The program will run without a printer (or even without HP-IL Module). Be sure your printer is set to the needed mode and assigned as the PRINTER IS before running the program. If it is available on your printer, be sure that automatic perforation skip is deactivated because this program advances paper between pages by printing blank lines, not by printing the formfeed character.

Paper should be loaded with the print head at the first line on the page. If a sheet-fed printer is being used see the section on the page length (^PL) command for instructions to have the program pause for you to load paper.

Running the Program

There is only one prompt at the beginning of the program.

```
report file:
```

Enter the name of a file in RAM or on the first (or only) mass storage device.

There is an optional second usage for this prompt; you may specify data to be interpreted before the file is read. This is helpful to enter page-control commands or mailing addresses without having to edit the file before each printing session. Any data which follows the syntax requirements of the program may be used provided it is separated from the file name by a space. The following example would use the file called "myfile" and begin the printing session by setting the page length, tab setting and printing current date before interpreting the file.

```
report file: myfile ^pl66 8 7 ^ta35^da ^sk1
```

Be sure not to specify a page length in this prompt if the file itself begins with a page length command or a blank page will be printed as the program interprets the second page length command.

If You Want to STOP

While the program runs there is little to do other than ponder the advancements being made in printer technology. If you watch the printout as it is progressing and see that it is doing something dreadful, press **ON** to suspend the program. If you have stopped the program in the middle of, for instance, several hundred line feeds, enter: END or ENDALL. This will not delete any files that were loaded, but will end the program and free the memory it uses. Also, if you entered this program from another program, the other program will not continue. Normally the program will continue reading your files (and displaying their catalog entries) then terminate properly. This method is only mentioned if you, for whatever reason, wish to terminate REPORT while it is still working.

Finished

When the report is completed the computer will display the number of words and the number of the last page printed. The word count includes any data separated by spaces excluding commands or any parameters required by commands. Two words separated by a hyphen or fixed space count as one word. The date, as entered with the date (^DA) command, counts as one word. Escape codes (^EC) and control codes (^CC) are not added to the word total. Copy mode counts as one word per line.

Any files which were loaded by the program during the run will be purged before the program ends.

If the program terminated with an error then the file, up to the last properly interpreted line, will be printed. Edit the file at that point and review commands or memory requirements it may have required to give you an indication of how to correct the error.

REPORT Program Commands

These commands are used to control how you would like REPORT to print your files. When the program is run it assumes some default values. You may change these values using commands in your files, or use them as is.

REPORT Default Settings

Page Length:	66	Left Margin:	10
Top Margin:	6	Right Margin:	70
Bottom Margin:	6	Fill Mode	
Single spacing		No page numbers	

```
AD  (ADvance to new page)
```

Stops printing on the current line. Advances the paper to begin printing on the next page. This function operates exactly like the Page Length command (PL).

```
CC  (Control Code)
    ^ccABC text...
```

Tells the computer to send the data following the command to the printer as printer control characters. These characters are assumed to be control codes that will not be printed and are therefore not added to line or word counts. All data up to the next space is interpreted as control codes. Any data or commands may follow the control codes on the same line if preceeded by a space. You may accumulate spaces with the fixed-space character (~).

```
CE  (CEnter on line)
    ^ce This is centered
```

Stops printing on the current line. Enters Center mode during which all of the data on each is printed centered. Center mode remains in effect until another mode command is used (FI,CO,PA). The text is centered as entered on the line, leading and trailing spaces are ignored.

```
CO  (COpy mode)
    ^co This will be copied on one line
```

Stops printing on current line. Enters copy mode during which the contents of each line is printed as written. No formatting is done. Other commands may be used while in copy mode if they preceed data on the line. You may, for instance, use use the date command (DA). Copy mode is preferred when printing information which is already formatted, such as when merging a

WorkBook71 spreadsheet which has been placed in a Text file, or for printer graphics.

DA (insert today's DAte)

Places the current date in the print out. The month is spelled out completely, followed by the day and year in numerical form. The date uses fixed spaces so that it will always be print complete on one line. Be sure to leave a space or punctuation after this function (or have it the last data on the line). No trailing space will be added. 09/04/86 is printed as:

September 4,1986

The addendum gives directions for changing to day,month, year and changing the spelling of the months.

DL (Distribution List) ^pL66 6 6^dL distfile:TAPE text...

Designates distribution list file name. A copy of the entire report will be printed for each line in the list. The distribution list command (DL) must be the first non-remark data in the file, and the file name must immediately follow the command. Specify a mass-storage device name with the file name if you wish to use a file which is not currently in RAM and you do not wish the program to load it. The distribution list file is a Text file with each line containing related data. The program uses one line from the file for each copy of the report.

LaFong Plumbing,123 Back St,Anaheim,CA,92802
Bissenette Market,789 Side St,Philadelphia,PA,34567

Each field is separated by a comma. This command is one of the few with restrictions to remember: You may use only one Distribution List per report, and the command must be the first non-remark data in the file. Other data or commands may follow on the line if a space separates the data from the distribution list file name. To recall data from any field in the distribution list use the Distribution Recall command (DR). In the example you can recall the zip code field with "^DR5". Another use for distribution lists is to print several identical copies of a report. Simply specify a Distribution List with as many lines in it as you need copies. Remember, it is not necessary to recall data from every field (or any field at all) in the assigned list.

```
DR  (Recall Distribution list field
    Dear ^dr1 ^paText...
```

The Distribution Recall command is used to recall a field from the current line in the Distribution List. The file must have already been assigned using the Distribution List command (DL). The same field may be used repeatedly or not at all in the report. You may not use Distribution Lists in a report that also uses the new file command (NF).

```
EC  (escape code)
    ^ecHC text...
```

Sends each of the following characters to the printer preceeded by the escape character (ASCII 27). In the example the printer would receive <esc> H and <esc> C. With many (non-HP) printers, this will turn on the Bold and underlined print mode. To send ESC SPACE use the fixed space character (~). HP printer commands often begin with "^EC&" followed by a space and the control code command (CC) and more data. Examples of commands for the HP ThinkJet printer are shown in the Addendum and later in this section in the discussion of the User function (U). As with other commands, data or commands may follow on the same line if separated from the escape string by a space.

```
FI  (fill mode)
    ^fi text...
```

Causes a printing break. Prints subsequent lines with as many whole words on each line as will fit within the margins. The left border is flush, the right border is not. Any extra spaces between words are deleted. Each time a word is followed by a period, comma, question mark or colon (.,?:) an extra space is inserted between it and the following word. This is the default mode REPORT assumes when it starts printing.

```
GR  (GRaph value)
    ^gr [1,2]/2
```

Builds a star graph using the results of the formula following the command. Does not cause a printing break. Data may preceed or follow the command on the same line. Syntax for using formulas is the same as for the WorkBook Formula command (WF). If the number of stars exceeds the remaining width of the current printed line then the stars will be truncated to fit on the line. This command is very useful with a label at the beginning of the line then tab to a specific column so that graphs on other lines begin at the same column for comparison. If data will fit then it will be added following

the stars (so that trailing labels may be used). A subsequent tab can continue data on the same line following the graph. Be sure to use the skip command (SK) to stop printing data after the graph and move to the next line.

```
^sk0 Row 1: ^ta10 ^gr[1,1] ^wf[1,1]
^sk  Row 2: ^ta   ^gr[1,2] ^wf[1,2]
^sk  Row 3: ^ta   ^gr[1,3] ^wf[1,3]
```

The example would begin each line with 'Row n:' followed by the graph and, to show that other data may follow, the actual number in the cell is placed after the graph on each line (using the WorkBook Formula command (WF)).

```
JU  (right-justify mode)
    ^ju text...
```

As with fill mode except spaces are added between words to make right border of the text a straight line. Will not justify the line completely if it requires a great number of fill spaces. The last line in a paragraph is not right justified. Proportional spacing is not supported. You may set your printer for proportional characters, but this program will not do microspacing.

```
MA  (set margins)
    ^ma10 70 text...
```

Set the margins to other than the default settings of 10 and 70. Lines may be up to 255 characters, this includes any control codes, so the effective maximum right margin is 132. The minimum left margin is one.

```
ME  (merge file)
    ^me flname This text will be printed before merging
```

Specify a Text file name. If other commands or data follow on the same line it must be separated from the file name by a space. Any data on the same line as the merge command will be read before beginning to read from the merged file. The program will suspend formatting the current text file and search for and merge data from a second file. Only one level of merge may be used. That is, you may not merge a file from a merged file. When the final data has been read the program will continue from the next line in the main text file. If you specify a device ID with the file name the program will read the file directly from Disc. Otherwise, if it is not in RAM, it will be searched for on the first (or only) Disc and loaded to the computer to be read from RAM, then purged when it has been read. Files found in RAM will be left there.

```
NF  (Assign a New main File)
    ^nf flname text...
```

Designates a new file to replace the main file. Remaining data on the same line is interpreted but data on following lines (if any) in the original file is ignored. The main use of this function is to link a number of files. You may not use new file in a report using the distribution list function (DL). Note the differences between New File (NF) and Merge (ME).

```
PA  (start a new paragraph)
    ^pa5 text...
```

Causes a paragraph break. The default number of spaces to indent is 5. If you wish, you may change the number of spaces to indent to any practical number. Using the paragraph command without any number following assumes you wish to indent the same number of spaces last used. A negative number of spaces WILL NOT cause a negative paragraph indentation.

```
PL  (set page length)
    ^pl 66 6 6 text...
```

The program will assume a page length of sixty-six and top and bottom margins of six each. The page length command is used to change these settings. This function causes a page break; that is, it will stop printing on the current page (if any data is on it) and begin a new page. It is a good idea to set the values at the beginning of the file. Any subsequent uses of Page Length will again cause the program to stop printing on the current page and start at the top of the next page.

Use a negative number for page length if you are using a sheet-fed printer and the computer will stop printing after each page and tell you when to insert paper. After changing paper press any key to continue printing.

```
PN  (Page Numbering)
```

Print page numbers. Pages are numbered from the beginning of the report. The page number command with any value, or none begins numbering, "^PN0" or "^PN-" stops printing numbers. You may optionally include a starting page number to use.

```
SK  (SKip number of lines
    ^sk1 Text and commands...
```

The default value is 1 or the value last used. "^SK0" means "skip zero lines", printing will halt on the current line and resume on the very next line. Using the skip command without any paramater will cause it to use the value you last used. The default is one line.

```
SP  (SPacing between lines)
    ^sp1 text and commands...
```

Does not cause a printing break. The program assumes you want all lines single-spaced. To change to double (or more) spacing use the skip command followed by, for instance, 2. There is no limit to the spacing imposed by the program, however, you will probably want to use a value of 5 or less. As with most commands, other data or commands may follow on the same line if separated by a space.

```
TA  (TAb)
    text ^ta40 text...
```

Does not cause a printing break unless the tab is to a position on the line which is already occupied. There is no default value; you must specify the number of spaces to indent the first time. The value last used is retained by the program. You may repeatedly tab to the same location on subsequent lines by entering the value on the first tab, then just specify "^TA" on following lines:

```
Hewlett-Packard                 ^ta52Richard E. Harvey
^sk0 1000 N.E. Circle Blvd.     ^taBox 5695
^skCorvallis, OR                ^taGlendale, AZ
^sk ta8 97330                   ^ta60 85312
```

Notice that the values to skip and tab were both specified only once until it is necessary to change them. However, if a command allows an optional parameter, it may not be followed by a number unless that parameter is also included. Any number would have been interpreted as the parameter for the command. This is illustrated above when using Zip codes where a paramater is specified, then a space before the Zip code.

```
WA  (assign WorkBook File)
    ^wa wbfile:tape
```

Data may be recalled from a WorkBook71 file using the "^WR" and "^WF" commands. Before you may recall data the program must know from which file to search for it. When the program comes upon the WorkBook Assign command it reads the current status from the specified file and tries to recall the Col and Row user labels (if the apropriate label file is available in

RAM). As you can see, it is important for the program to know this data before trying to recall individual cells.

```
WR  (recall value from WorBook)
    ^wr [col,row] text and commands...
```

Specific cell contents may recalled from an active WorkBook file by specifying first the column then row coordinate of that cell. You may specify user or default labels, with or without [brackets]. Text data from the file is recalled as written. Numbers and results of formulas are recalled as formatted in the file. Blank cells are recalled as the null string, not Ø.

```
WF  (calculate formula, return results)
    ^wf 2*sqr[2,3]  Text...
```

This is much the same as entering data into a results only (=) WorkBook cell. The formula you specify is calculated then the results entered into the report in the Global display format of the WorkBook file. All data following the WorkBook Formula command and up to a following space is interpreted as a formula. It is not necessary to have an assigned WorkBook file if the formula does not refrence any cells in the file.

```
ZZ  (fill line with repeating characters)
    ^zz x text and commands...
```

Causes a printing break. Accepts any parameter you specify up to the width of one printed line (in which case it wouldn't repeat at all). If no parameter follows, the program will print a full line of dashes "-". You may specify a repeating string. To embed spaces in your string use the fixed space character "␣". A space designates the end of your repeating pattern. Any data or command may follow on the same line.

```
U_  (User function)
    ^uEXP text...
```

This function is versatile and unstructured and may, therefore, be confusing at first. The function allows you to call a SUB program to add data or alter data being read from the current file. This type of operation is often called a "macro".

What happens is that when the program comes upon the command it trys to call a SUB program with the name specified. The program is called while passing a variable which contains whatever data remains on the same Text file line as the command. When the SUB ends then this program continues by reading whatever was passed back to the program in that string

variable.

Let's start with a simple example. Suppose you want a function to set a ThinkJet printer in compressed pitch mode; this is done with the sequence CHR$(27)&"&k2S". This can be done in the line by using the escape code function (EC) followed by the control code function (CC) and that strange string:

```
^EC& ^CCk2S
```

There are many ways to do the same thing with a SUB. Let's take an easy method, though probably not the most memory efficient.

```
10 SUB COMP(E$)
20 E$='^CC'&CHR$(27)&'k2S '&E$ @ END SUB
```

The sub enters a control code command followed by the actual data then a space. The original string is then added to the output in case it already contained some data. Since a string is furnished with the remaining data on the same line we could do anything with that string. We'll write three other functions, One deletes the remainder of the string (that is, the remainder of the current line) if the time of day is before noon; the second one uses the WBLEX function HI$ to set the high bit (add 128 to the NUM) on each character. The third is like the second but only alters uses the first word in the string to demonstrate the usage of other keywords in WBLEX.

```
 50 SUB NOON(S$) @ IF TIME<43200 THEN S$=""
 60 END SUB
200 SUB HI(S$) @ S$=HI$(S$) @ END SUB
300 SUB HI1(S$) @
    S$=HI$(NXTWORD$(S$))&WTRIM$(S$,NXTWORD$(S$))
```

As you can see, anything can be done with the string which will eventually supply a single string result at the end of the SUB program is permissable. The maximum length of the string (including the part passed to the SUB) is 255 characters. If there is a possibility of an error occuring in the SUB be sure to include an error trap (ON ERROR) so that it will terminate properly and return control to REPORT without halting for the error. The SUB must be in RAM, REPORT will not load it. As with other SUB programs, any number may be in a file.

There is no limit to how you may use the user function. Imagine using the remaining on the line as a prompt for input, perform an operation on that value, then return the results in the string. The current files remain assigned when calling the sub; you may (carefully) recall other data from the file. Be cautious about recalling data from the Main, Merge, or Distribution List files because it will alter the file pointer and affect the next line to be read when REPORT continues.

This can be to your advantage if your SUB wishes to skip several lines in the Main file if certain conditions are met. Do not close any of these files or the program will terminate the next time it tries to read the file. The (possible) open files are as follows:

#1	Main Text file
#2	Merge Text file
#4	Distribution List
#5	WorkBook71 Data file
#6	WorkBook71 Label file

A sample MACRO file is furnished on the Disc; it is called (conveniently enough) "MACRO". It contains some utilities which may be used with a ThinkJet printer.

NORM	- set normal pitch
EXP	- expanded pitch
COMP	- compressed
BOLD	- bold pitch
NOBOLD	- bold off
SPACE8	- close spacing
SPACE6	- normal spacing
TIME	- the time

Using WorkBook71 Files with your programs

This section discusses some of the design philosophy and structure of the WorkBook file. It is not necessary to read or understand this section to use the programs provided in this package. This data is provided on an as-is basis and is not intended to be a tutorial. It cannot be overstated that casually altering the data stored in the files can make them unreadable by the WorkBook editor. It is suggested that data be read only from these files. Write data to the files only if no other methods can be found.

More information about programming the HP-71 can be found in the book "The BASIC HP-71" by this author.

An annotated listing of the FINDER program is provided as an example of usage of these files. Since a program is a living thing, the listings may not reflect the actual code of your copy of the program.

WorkBook File Structure

The DATA file used by the WorkBook system is a standard file as written and read by the HP-71 Basic Operating System. The size of the file is fixed at the time of creation. This is done because of the need for random read/write, because of the necessity of a very large LEX file for other methods, and because this format allows the creation of files on mass storage that can be randomly written to and read.

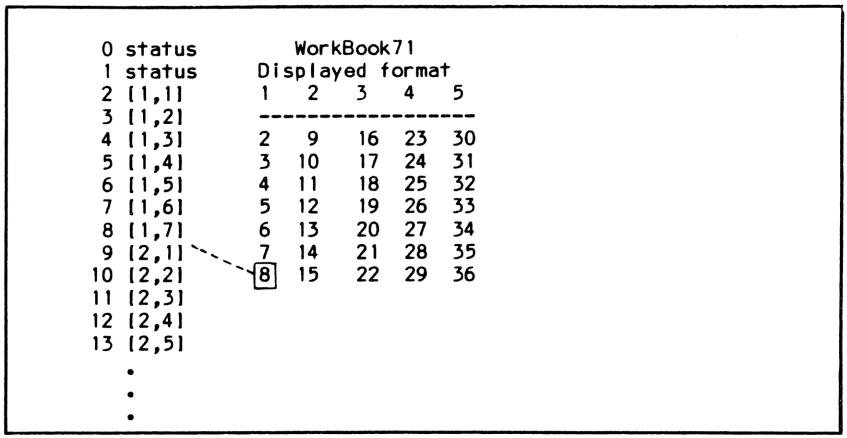
There is some information about the file which must be maintained within the file for proper access. This information includes the number of columns and the location of the cell edited the last time the file was accessed. The length of records in the file is set by the user. If the needed information had been stored one piece per record it would consume 11 records, a considerable waste if the record length is quite long. It was determined that the minimum usable length for a record was about 10 characters so the data was packed to fit within that restriction for less waste. The file information is in 10 bytes each on record 0 and record 1 of the file.

File status includes length of each record, name of the header file, status of the user flags 0-3, the global format, number of cols and rows in the file, the coordinate of the last cell edited and the coordinate of the cell in the upper left corner of the display. They are stored in packed form, one or two bytes representing each piece of information. The simplest byte is the status of flags 0-3, a hexadecimal number. The coordinate of last col and last row edited is the other extreme. The largest number of cells possible is 64K cells and can be represented by two bytes each for col and row. The coordinate of the last cell edited is probably only useful for the WorkBook editor itself so the data is packed again. Using a formula listed below, the actual location in the file of the last cell edited is stored as two bytes, then, using the number of cols and rows in the file, the current row then current col are extracted from these two bytes. The method for turning two bytes into a coordinate using UNPAK is demonstrated on lines 5000 and 5005. The function PAK$ has the opposite effect. It is fairly simple to extract this data, and it is only done when the file is first assigned.

Files are dimensioned to a specific format when created. When created cells are filled with a byte designating cell format and a filler space before the file is ready for editing. There is no restriction on the number of cols and rows but, because of the limit imposed by DATA files, the total number of cells cannot exceed 65535 (including

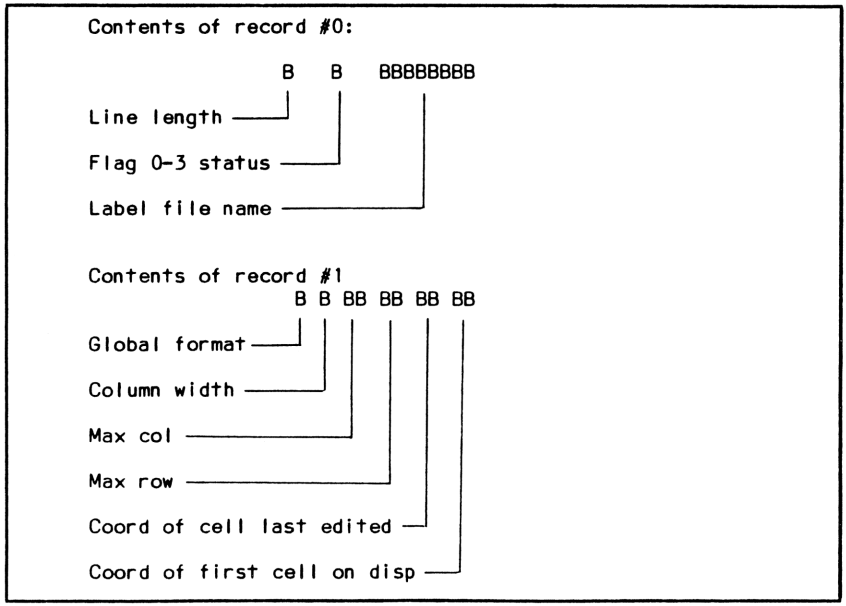
record 0,1) and the minimum line length is represented as 9 characters. The actual minimum line length is 10 characters because the first byte in each cell contains the individual cell format. Each cell has an additional three bytes used by the Operating System for data about the cell (record). The first byte and these three system bytes are automatically accounted for when the file is created.

A hypothetical file is represented below. It contains 5 cols and 7 rows. On the left is how the data is actually stored in the DATA file. Note that the cell coordinates are not actually stored in the file, but are computed as needed using a formula listed later. Numbers in the cells relate to their actual location in the file.



Record Format

As mentioned earlier, records 0 and 1 in each file contain information about the physical makeup of the file.



Format of Label File
The header file is separate from the main WorkBook file because of the fixed format of the main file and to keep memory usage to a minimum. The file is first read to find the maximum length of the strings. Then the strings are dimensioned to the number of labels*6 and the labels are read into the strings. It is faster to use the POS function on one long string than to sequentially read separate strings so this format is much preferred.

Each label begins with a leading space then 5 characters. Labels must begin with ASCII's above 48, otherwise programs will use default labels. Upper- and lower-case are allowed. The leading space is used to separate the labels and avoid the possibility of finding part of one label and part of the next when actually searching for one later in the string.

```
Label file:
n = number of col labels
n = number of row labels
$ = actual col label string
$ = actual row label string
```

An example of reading this file type is demonstrated in the FINDER program on lines 5020 and 5030. First the two variables representing number of cols and rows are read. Then the strings are DIMmed to six times the number of labels (because of the label plus leading space). Finally the contents of the two label strings are read to the newly DIMensioned strings.

Formulas for Finding Data
Often it is easier to create a user function than to write a complicated formula several times. Functions to read and write data and some simplified code examples follow.

Decode the two-byte numbers in records 0 and 1 with the following or the UNPAK function in WBLEX.

```
DEF FND(Q$)=HTD(DTH$(NUM(Q$))[4]&DTH$(NUM(Q$[2]))[4])
```

USER FN to encode two bytes to the above format. The same as PAK$ in WBLEX. Note a maximum value of 65535:

```
DEF FNH$(Q)= CHR$(HTD(DTH$(Q)[2,3])) &CHR$(HTD(DTH$(Q)[4]))
```

To recall a coordinate from a file when COL(C), ROW(R), MAXCOL(C9) and MAXROW(R9) are known:

```
READ #1,C*R9-R9+R+1;Q$
```

We can find COL and ROW when all that are known are the MAXCOL(C9), MAXROW(R9) and the coordinate(n). This is done in two steps. First we find the current COL(C):

```
C=CEIL((n-1)/R9)
```

Then, using the COL we found above, we find the ROW:

```
R=n-(R9*(C-1))-1
```

An example of the usage of the above formulas is to recall the COL and ROW that had been last edited when the file was last read:

```
READ # 1,1;Q$    ! read the line
C9=FND(Q$[3,4]) ! find MAXCOL
R9=FND(Q$[5,6]) ! find MAXROW
Q=FND(Q$[7,8])  ! find last coord
C=CEIL((Q-1)/R9) ! decode the COL
R=Q-(R9*(C-1))-1 ! decode the ROW
```

We now have: C9=MAXCOL, R9=MAXROW, C=CURRENT COL, R=CURRENT ROW.

The Format Byte

Byte 1 of record 1 in the file and the first byte in each cell are reserved for the format (FIX setting). These values are found simply by reading the appropriate data then using NUM(Q$). If the value is between 0 and 9 then this is the FIX notation. Values of 10 or larger indicate that the global format is to be used.

When files are created, the Global format is set for 2 and each cell is set to 10. Thus, unless individually changed, cells reflect the global format.

Cell Contents

Bytes two through the maximum length allowed by the file configuration are stored as represented to the user. Byte 2 also designates the cell type. If the byte is "=" then the cell contains a number only. If the byte is "+" then there should be a number, followed by a space and "@" and finally a formula. Any other value for this byte designates the cell as a non-numeric data cell and it is interpreted as a value of zero during recalculation.

```
 10 CALL FINDER @ SUB FINDER !            WorkBook71 search program
 11 ! Copyright (c)1986 Richard E. Harvey
 15 DIM S$[96],E$[96],G$[96],K$[256],Q$[96],X$[96]
 20 INTEGER S(12),E(12),G(12),R,E1,S1,G1,C,L,C1,C8,X,Y,Z
 30 CLFLS @ ON ERROR GOTO 'N' !              close any open files
 40 'N': E(0)=1 @ W=1 @ INPUT 'search file:';W$ ! get the file name
 45 IF NOT LEN(W$) THEN CAT ALL @ GOTO 'N' !    CAT ALL if no name
 50 CALL INCAT(W$,W) @ IF W=3 THEN 57 ELSE IF W THEN 'N'
 52 DISP 'Load/Virtual' @ ON POS('LV',WTKEY$) GOTO 54,56
 54 COPY :MASSMEM TO W$ @ GOTO 57 !                  load the file
 56 W$=W$&':MASSMEM' !                               virtual file
 57 ASSIGN #1 TO W$ @ GOSUB 5000 @ ON ERROR GOTO 60 ! open the file
 60 DISP CHR$(27)&'<'; !                          turn off the cursor
 65 INPUT 'output file:';D$ @ CALL INCAT(D$,Z) @ IF Z>1 THEN 60
 70 IF NOT Z THEN CREATE TEXT D$ !                   a new output file
 80 ASSIGN #2 TO D$ @ RESTORE #2,9999 !       pointer to end of file
 90 LINPUT 'field separator:','°';Z$
100 INPUT 'output rows:','1,'&STR$(R9);S$,E$ !     copy which rows?
105 R1=NVAL(S$) !                  transform into default row numbers
110 IF NOT R1 THEN R1=IP((POS(UPRC$(R$),' '&UPRC$(S$))+5)/6)
112 R8=NVAL(E$)
114 IF NOT R8 THEN R8=IP((POS(UPRC$(R$),' '&UPRC$(E$))+5)/6)
118 IF R1*R8<1 OR R1>R9 OR R8>R9 THEN 110 !       good row numbers?
130 M=2 @ X$='search' @ GOSUB 8000 @ S$=E$ @ S1=E1 ! search strings
135 IF NOT S1 THEN 160 !   S1 is true if we have any search strings
140 FOR L=1 TO S1 @ S(L)=E(L) @ NEXT L @ IF S1=1 THEN 160
150 DISP 'strings mandatory Y/N' !       must all strings be found?
155 M=POS('YN',WTKEY$) @ IF NOT M THEN 155 !    wait for a good key
160 X$='< or >' @ GOSUB 8000 @ G$=E$ @ G1=E1 @ IF NOT G1 THEN 180
170 FOR L=1 TO G1 @ G(L)=E(L) @ NEXT L
```

```
 180 X$="exception" @ GOSUB 8000 @ DISP HGL$(TIME$)&' '; @ OFF ERROR
 500 DESTROY R$,H$,W$ @ INTEGER F(C9) !     trash unneeded variables
 510 IF S1 THEN 530 !                    not S1 means no search strings
 520 FOR C=1 TO C9 @ F(C)=1 @ NEXT C @ GOTO 1000 !    flag each cell
 530 FOR L=1 TO S1 @ K$=S$[L*8-7,L*8] @ R=S(L)
 540 GOSUB 8500 @ FOR C=1 TO C9 !                     do comparisons
 550 GOSUB 9000 @ IF POS("/"&UPRC$(Q$)&"/",K$) THEN F(C)=F(C)+1
 560 NEXT C @ NEXT L
1000 IF NOT E1 THEN 1500 !          not E1 means no exception strings
1010 FOR L=1 TO E1 @ K$=E$[L*8-7,L*8] @ R=E(L)
1020 GOSUB 9000 @ IF POS("/"&UPRC$(Q$)&"/",K$) THEN F(C)=0
1025 GOSUB 9000 @ IF POS("/"&UPRC$(Q$)&"/",K$) THEN F(C)=0
1030 NEXT C @ NEXT L
1500 DESTROY E1,E$ @ IF NOT G1 THEN 2000 !         < or > comparisons
1510 FOR L=1 TO G1 @ K$=G$[L*8-6,L*8] @ R=G(L)
1520 Z=POS("<>",G$[L*8-7,L*8-7]) @ IF NOT Z THEN 1590
1530 GOSUB 8500 @ FOR C=1 TO C9 @ GOSUB 9000
1535 Q$=UPRC$(Q$) @ ON ERROR GOTO 1580
1540 IF LEN(Q$)>LEN(K$) THEN Q$=Q$[1,LEN(K$)] !    equal len strings
1550 IF LEN(K$)>LEN(Q$) THEN K$=K$[1,LEN(Q$)]
1560 IF Z=1 AND Q$>K$ THEN F(C)=0
1570 IF Z#1 AND Q$<K$ THEN F(C)=0
1580 NEXT C
1590 NEXT L
2000 DESTROY G1,G$ @ IF M#1 THEN 2500 !          mandatory qualifiers
2010 FOR C=1 TO C9 @ IF F(C)#S1 THEN F(C)=0
2020 NEXT C
2500 FOR C=1 TO C9 @ IF NOT F(C) THEN 2530 !     data to output file
2510 K$="" @ FOR R=R1 TO MIN(R9,R8)
2515 GOSUB 9000 @ ON ERROR GOTO 2530
2520 K$=K$&Q$&Z$ @ NEXT R
2525 PRINT #2;K$[1,LEN(K$)-LEN(Z$)]
2530 NEXT C
3000 CLFLS @ BEEP @ DISP TIME$ @ END !                                bye
5000 READ #1,0;H$ @ H$=H$[3] @ READ #1,1;X$ !  read status from file
5005 C9=UNPAK(X$[3]) @ R9=UNPAK(X$[5])
5010 H=1 @ CALL INCAT(H$,H) @ IF H#2 THEN H=0 @ RETURN
5020 ASSIGN #3 TO H$ @ READ #3;C2,R2 @ DIM R$[MAX(C2*6,R2*6)]
5030 READ #3;R$,R$ @ RETURN
8000 FOR L=1 TO 12 !                  input loop used by all routines
8010 DISP X$&" str#";L; @ LINPUT ":";K$ !                the prompt
8020 IF NOT LEN(K$) AND L=1 THEN E1=0 @ RETURN !        user is done
8030 IF NOT LEN(K$) THEN E1=L-1 @ RETURN
8040 E$[L*8-7,L*8]=UPRC$(K$) !            move data to output string
8050 K$=STR$(E(L-1)) @ ON ERROR GOTO 8060
8060 INPUT "row:",K$;K$ !                 where are we going to look?
8062 R=NVAL(K$) !                          make sure it is a good row
8064 IF NOT R THEN R=IP((POS(UPRC$(R$),' '&UPRC$(K$))+5)/6)
8066 IF R<1 OR R>R9 THEN 8060 !          a bad row, ask for a new one
8070 E(L)=R @ NEXT L @ E1=12 @ RETURN !             end of input loop
8500 K$=REV$(WTRIM$(REV$(K$))) @ RETURN !       trim trailing spaces
9000 READ #1,C*R9-R9+R+1;Q$ @ Q$=Q$[2] !     read a cell from source
9005 IF POS('+=',Q$[1,1]) THEN Q$=CELVAL$(Q$) !     is it a += cell?
9010 RETURN
9600 SUB INCAT(F$,T) !                              the INCAT subprogram
9610 ON ERROR GOTO 9670 @ DISP CHR$(27)&'>'; @ CAT F$ !  display cat
9630 T=POS('TESDDABILEKEBAFO',WTRIM$(DISP$[12])[1,2]) !    file type
9635 IF NOT MOD(T,2) THEN T=20 ELSE T=(T+1) DIV 2 !    into a number
9640 END !                                                        bye
9670 T=21 @ IF ERRN=57 THEN T=0 !          0= no file,  21= bad name
```

ADDENDUM

Configuring WorkBook Editors for Monitors
As furnished, the WorkBook Editor (WB) and the Text Editor (TED) are configured for use with a 32 column, 16 line video interface. These programs may be adapted for use with other display sizes by editing the program. The first line in both programs, line 10, supplies the parameters for calling the program. The examples show the WorkBook editor, but the method is the same for TED.

```
10 CALL WB71(32,16)
```

The numbers in parentheses represent first the number of characters per line then the number of rows. To use, for example, a 40 character by 24-row display device change line 10 to:

```
10 CALL WB71(40,24)
```

Be sure to save an unaltered copy under a similar, but slightly different file name before performing this surgery.

It is also possible to CALL the programs for use with a different display size without altering them by entering, from the keyboard:

```
CALL WB71(C,R)
```

Where C is the number of characters per line and R is the number of rows. You may use actual numbers, variable names (such as C) or mathematical expressions to designate number of cols and rows. This is helpful if you normally use one size interface and wish to one time use another without mucking about. Remember that the file names are WB and TED while the SUBs within them are called WB71 and TED71. No other lines in the programs should be altered. In fact the programs can be made unusable through casual editing, so please don't.

HP DISPLAY ESCAPE CODES
The escape sequences used by this system are "standard" HP codes. Some terminals or other brands of computers operating as terminals respond differently to these escape sequences. Check your terminal's manual for methods to map these codes to those the terminal can handle. In fact, before even purchasing the terminal, it is a good idea to make sure it can handle HP escape sequences.

COMPATIBILITY CAUTION
LEX files in RAM or ROM which take over the machine to provide new operations such as constant clock display, appointment mode or operating system shell may prove unreliable when used with this package. No compatibility warranty is implied or stated.

WBLEX KEYWORD CAUTIONS
Programs in this package are for use with WBLEX version "WB:D", do not attempt to use them with "WB:C" or earlier. When new keywords were added to the file the IDs of a few other keywords were altered to make room for them.

If you have written programs using an earlier version of WBLEX they may be adapted for use with this version. Purge WBLEX "WB:D" from RAM then load the program you wish to update and WBLEX version "WB:C" (or "WB:B") to RAM, then enter:

```
TRANSFORM filename INTO TEXT
```

If the program uses the keyword "HGL$" then you will have to edit the line (with a Text Editor) and change it to "HI$". Now, purge WBLEX "WB:C" from the computer and load "WB:D" then enter the following from the keyboard:

```
TRANSFORM filename INTO BASIC
```

It is suggested that if you have an earlier version of WBLEX that you discard it once your programs have been modified for the later version.

REVISIONS

The VER$ response of the WBLEX LEX file represents the current version of the package and does not necessarily represent an updated version of the LEX file itself. The first release was "WB:B". TED (the Text Editor) was added as well as a number of revisions to other programs (including WBLEX - to recognize external keyboards and more display devices) in "WB:C". "WB:D" brought with it the SORTWB and LED programs as well as doubling the speed of REPORT and adding some new keywords.

As with any programming project, changes will continue to be made. Please write including the serial number on the first page of this manual and the VER$ of the LEX file if you would like to be advised of upgrades. New revisions and added program modules will be available at minimal cost. Feel free to offer suggestions and "wish lists".

INVERSE VIDEO

When used with a monitor the WorkBook editor (WB) displays the Col and Row labels in inverse video (black on white). Prompts for a single keystroke, such as "Y/N" which are displayed in uppercase to represent a keystroke are also displayed in inverse video. The HP-71 usually displays these characters as normal characters on the built-in LCD. The computer allows you to create an alternate character set which will be displayed at these times. A BASIC program is furnished on the Disc called CHARSET. Its two options are underlined and inverse characters. RUN or CALL the program and press the appropriate key (UI), the program will create the character set from the display. The program works by displaying 8 characters at a time then reading their bit patterns and modifying them.

Once the alternate character set is enabled you can use it in any program, or even assign the characters to keys. The WBLEX LEX file function HI$ makes it easy to display these characters. To assign the characters to keys use:

```
DEFKEY '?',HI$('?');
```

The alternate character set occupies 768 bytes of RAM. This character set may be eliminated, and the memory consumed regained by entering:

```
CHARSET ''
```

REPORT Date Format

Month, day, year and each month fully spelled in English is the default method for printing the date with the DA command. REPORT can be modified to print the date in other forms. Edit the program. The first line designates the program as a SUB and CALLs it. The second and third lines are the DATA statements which tell the program how to print the date. Line 105 begins with the DATA statement then MDY for "Month Day,Year". You may change to listing "Day Month,Year" by

changing the MDY to DMY. As you will also see when you edit line 105, the first six months of the year follow MDY on the same line. The last six months are listed on line 106. You may change the spelling as you would like providing you separate each month with a comma. Again, please don't edit other lines in the program. Remember to keep an unaltered copy of the program under a similar but slightly different name.

ThinkJet Printer Codes

Many printers use a single escape sequence to change printing modes. The REPORT "EC" command makes it easy to use these sequences. HP printers often use a sequence of codes which begin with the escape code. If you had used "EC" it would have sent CHR$(27) with each character. The correct method is to use "EC" and the first character to send the escape, then "CC" to send the body of the command. Below is a list of HP printer codes. l is lowercase L.

```
Normal Spacing:       ^EC& ^CCl6D
Close Spacing:        ^EC& ^CCl8D
Compressed:           ^EC& ^CCk2S
Expanded Compressed:  ^EC& ^CCk3S
Normal Type:          ^EC& ^CCk0S
```

ATTN KEY

A very rare occurance can happen if either the TED or the WB program was terminated by an HPIL error such as a Disc Drive failure. The ATTN key may have been disabled and will no longer function correctly. This key can be restored by either INIT-1 or by carefully entering the following:

```
POKE '2F441','0'
```

IF YOU NEED HELP

This manual is as complete as possible but, as is always the case, some things could always stand further explanation. Should you need help, or find something that just won't seem to work for you, feel free to write to the address below. Be sure to include printout or Disc/tape to help explain the problem. Please remember that profit on this package is (at best) minimal; it is beyond the resources of the author to provide extensive free customizations.

Quantity discounts and site licenses as well as customized programs are available.

Other software is or will be available for the HP-71 and HP-75 Portable Computers. Contact the author for details or if you have any questions.

Richard E. Harvey
Box 5695
Glendale, Arizona 85312 USA

GLOSSARY

A glossary of terms is available in the HP-71 Reference Manual beginning on page 346. Some other terms are used in this manual which are not described in that glossary.

BROWSE LEVEL An operational mode in the WorkBook Editor which allows inspection of data in a file without the possibility of accidently altering it. Often called top level.

CELL The intersection of a column and row in which data is stored. Represented as a record in in a Data file.

CELL TYPE Refers to the first displayed character in a cell. If it is an equals (=) then the cell contains a number. If the character is a plus (+) then the cell contains a number and the formula from which the number was computed. Any other first character means the cell is a general data cell.

CIRCULAR REFERENCE A formula contained in a cell which directly or indirectly (by referring to another cell which in turn refers to this cell) refrences itself. Since the value will change whenever the cell is recalculated, a reliable value can not be obtained.

COL Abreviation for the word COLUMN.

COLUMN The vertical arrangement of cells in a Worksheet.

COMMAND A keystroke or series of keystrokes which tell the computer to perform an operation (see command level). Also a sequence of characters preceeded by a circumflex (^) in a file which are interpreted by a formatter program.

COMMAND LEVEL An operational mode in the WorkBook Editor in which some edit keys perform special operations and cell editing is disabled.

COPYRIGHT A benevolent set of laws which allow the fruit of one's labor to remain one's own. Through respect of these laws comes the incentive to create new products and maintain those which are extant.

COORDINATE The cell refrence specified by a col and row.

DEFAULT Values which are supplied by a program if not specified by the user. For instance, global col width defaults to 8.

EDIT LEVEL An operational mode in the WorkBook Editor in which the entire keyboard is active. Horizontal arrow keys move the cursor within the current line. Vertical arrow keys move from cell to cell.

FIELD An entry in a Data Base such as Name or City. Generally fields are defined by rows in WorkBook file.

FILE MANAGER A simplified form of Data Base Management system.

GLOBAL A value which represents status affecting an entire file. Such as global COL Width or global format.

FORMATTER A program which reads a file and prints a doccument using data and commands within that file for instructions.

HEADER Name given a col or row label by many spreadsheet programs.

HPAF Hewlett Packard Applications Format. A type of DATA file created by some HP programs. Uses the first three records for file status and multiple data items per record.

LABEL A refrence to either a col or row. May be either a numerical value or a series of user defigned string. Some Spreadsheet programs call a label a header.

MAXCOL Refers to the largest-numbered col in a file.

MAXROW Refers to the largest-numbered row in a file.

OUTPUT FILE A file to which the results of a requested operation are directed. Such as the file used by the PORTER program to store the results of a search.

ROW The horizontal range of cells in a Worksheet. Also a field in a Data Base.

SEARCH STRING Data which is specified to be compared for equality with data in a file. Such as the FETCH command in the WorkBook Editor or data input to the FINDER program.

SOURCE FILE The file from which data is to be found for comparisons or transforming into another data format.

SPREADSHEET A program in which data is represented in a tabular form of columns and rows. A file edited by a Spreadsheet program is sometimes also called a Spreadsheet.

STORE A simplified term for saving data to a file or the results of input to a variable. The term is sometimes inapropriately used but should be comfortable for users of HP calculators.

TEXT A cell in a spreadsheet which is either empty or contains non-numeric data. Cell contents which do not begin with + or = as the first character.

VIRTUAL FILE A file which is resident on Disc or Cassette and accessed there without first being loaded to RAM.

WORKSHEET A Spreadsheet. A term which attempts to designate a multiple-purpose Spreadsheet file.