C71 Calculator
Owner's Manual
For the HP-71B

Corvallis Microtechnology, Inc.
895 N.W. Grant Ave.
Corvallis, OR 97330

CMT

CMT-1000

C71 Calculator

Owner's Manual

For the HP-71

July 1987

Notice

Corvallis Micro Technology and Iota Systems makes no express or implied warranty with regard to the keystroke procedures and program material offered or their merchantability or their fitness for any particular purpose. The keystroke procedures and program material are made available solely on an "as is" basis, and the entire risk as to their quality and performance is with the user. Should the keystroke procedures or program material prove defective, the user (and not Corvallis Micro Technology, Iota Systems, nor any other party) shall bear the entire cost of all necessary correction and all incidental or consequential damages. Corvallis Micro Technology or Iota Systems shall not be liable for any incidental or consequential damages in connection with or arising out of the furnishing, use or performance of the keystroke procedures or program material.

We gratefully acknowledge and express our thanks to Hewlett-Packard for allowing us to reproduce sections of the following manuals:

HP-11C Owner's Manual
Copyright Hewlett-Packard 1984
Reproduced With Permission

HP-12C Owner's Manual
Copyright Hewlett-Packard 1982
Reproduced With Permission

HP-15C Owner's Manual
Copyright Hewlett-Packard 1984
Reproduced With Permission

HP-16C Owner's Manual
Copyright Hewlett-Packard 1984
Reproduced With Permission

HP-41 Advantage Manual
Copyright Hewlett-Packard 1985
Reproduced With Permission

Contents

Section 1
Getting Started

What C71 Does

The C71 module contains a program called C71 that allows your HP-71 to act like a Hewlett-Packard Series 10 calculator. C71 simulates many of the functions found on the HP-11C (Scientific Calculator), the HP-12C (Financial Calculator), the HP-16C (Computer Scientist), and the HP-15C (Advanced Scientific Calculator). With just a few key strokes you can switch from Basic into the calculator and easily enter your calculations.

Installing and Removing the C71 Calculator Module

The C71 Calculator module can be plugged into any of the four ports on the front of the computer.

****************************** CAUTION ********************************

* Do not place fingers, tools, or other foreign objects into any of the ports. Damage to plug-in module contacts and computer's internal circuitry may result. Such actions could result in minor electrical shock hazard and interference with pacemaker devices worn by some people.

* Be sure to turn off the HP-71 (press **[f][OFF]**) before installing or removing a plug-in module.

* If a module jams when inserted into a port, it may be upside down. Attempting to force it further may result in damage to the computer or the module.

* Handle the plug-in modules very carefully while they are out of the computer. Do not insert any objects in the module connector socket. Always keep a blank module in the computer port when a module is not installed. Failure to observe these cautions may result in damage to the module or the computer.

****

To insert the module, orient it so that the label is right-side up, hold the computer with the keyboard facing up and push in the module until it snaps into place. Be sure to observe the precautions described above.

To remove the module, grasp the lip on the bottom of the front edge of the module and pull the module straight out of the port. Install a blank module in the port to protect the contacts.

Starting C71

To enter the calculator type:

C71 [END LINE]

the display will clear and then a number and the mode will be displayed. The number will be the value that was displayed the last time you used C71 or 0 if this is the first time you have used C71.

Leaving C71

There are two methods for leaving C71: [f][OFF] and [g][EXIT]

Turning the HP-71 Off

Pressing [f][OFF] will turn the HP-71 off, but will leave the calculator active. The next time **ON** is pressed, C71 will be reactivated and you will be working in the calculator again. The calculators status is saved prior to turning off and restored upon turning back on. However, if something should happen so the calculator does not come back on, because the status was saved, typing **C71 [END LINE]** will restart the calculator just where you left it. For more information on the interaction between the calculator and the HP-71 BASIC refer to Appendix D: HP-71 System Usage.

Returning to BASIC

Pressing [g][EXIT] will store the current status of C71 and return control of the HP-71 to the BASIC operating system. Pressing **C71 [END LINE]** will restart the calculator just where you left it.

Display

Mode

C71 has two basic modes of operation which coincide with the two overlays provided for the keyboard. The first mode is *Engineering/ Business* mode. When C71 is in this mode an **:F** will be displayed on the right side of the display. This mode uses floating point numbers. Most operations such as financial, trigonometric, and engineering calculations are done in this mode. In this mode many of the functions found on the HP-11C, HP-12C and HP-15C calculators are simulated.

The second mode of operation is the *Computer Scientist* mode which simulates many of the functions found on the HP-16C. C71 signifies this mode by displaying a **:H**, **:O**, **:D**, or **:B** on the right hand side of the display. These letters signify the various number base formats (hexadecimal, octal, decimal, and binary respectively) available in the Computer Scientist mode. Computer Scientist mode operates only on integers. Refer to the "Number Base Modes" in section 10 for further information.

To switch between modes press [f][RUN]. The display will switch between the **:F** and **:H**, **:O**, **:D**, or **:B** to signify which mode the software is in.

While in Engineering/Business mode, you can choose a format in which to display the numbers. They are standard, fixed, scientific and engineering formats. Refer to section 3, Display Control, for further information.

Flags and Annunciators

Your HP-71B display contains five flags (0 through 4) on the right hand side and several annunciators on both the right and the left hand side. These flags and annunciators tell you the status of the computer during certain operations. They are described, with the operations they refer to, in the appropriate sections of this manual.

Flag	Definition
0	Begin mode (Finance)
1	M.DY mode (Calendar)
2	Leading zeros (Computer Scientist)
3	Carry (Computer Scientist)
4	Out-of-Range (Computer Scientist)

Negative Numbers

To make a displayed number negative--either one that has just been keyed in or one that has resulted from a calculation--simply press [CHS] (*change sign*). When the display shows a negative number--that is, the number is preceded by a minus sign--pressing [CHS] removes the minus sign from the display, making the number positive.

Display Clearing: [CLRx] and [<]

C71 has two types of display clearing operations, [CLRx] (*clear X*) and [<] (*back arrow*). When [CLRx] is pressed, any displayed number is cleared to zero.

Pressing [<] after executing almost any function clears all digits in the display to zero.

Keystrokes	Display		
12	**12**	**:F**	
[f][1/x]	**.0833**	**:F**	
[<]	**.0000**	**:F**	Pressing [<] after executing a function clears all digits in the display to zero.

After keying in a new number, if you press [<] before executing a function (that is, before terminating digit entry) the last digit you keyed in is deleted. After you delete one or more digits, you can, if you want, key in new digits to replace them.

Keystrokes	Display		
12345	**12345**	**:F**	
[<]	**1234**	**:F**	
[<]	**123**	**:F**	When digit entry has not been terminated, [<] acts on each digit separately.
9	**1239**	**:F**	

Scrolling

In Engineering/Business mode the display does not scroll. The display may be scrolled in Computer Scientist mode. For further information on scrolling in Computer Scientist mode refer to section 10.

Overflow and Underflow

C71 uses the HP-71's operating system and therefore has the same overflow and underflow conditions as the BASIC system does. The way C71 handles an overflow or underflow depends on the *DEFAULT* settings of the IEEE traps. On a **Memory Lost** or initial power on condition the HP-71 sets **DEFAULT ON**. With **DEFAULT ON** overflows are set to the maximum magnitude value of the machine: 9.99999999999E499. On underflow it defaults to 0. Whenever an overflow or underflow occurs the calculator will beep (unless the beep has been disabled while in the BASIC system). For more information on the behavior of the HP-71 with other DEFAULT settings, on the HP-71's IEEE math handling, and on the HP-71's numerical range, refer to section 2 of the HP-71 Owner's manual and to "The IEEE Proposal for Handling Math Exceptions" in the HP-71 Reference manual.

Working

When C71 is calculating a result that takes several seconds, the display will show or flash:

Working...

to let you know that it is working on your calculation. The functions that will display **Working...** are [i], [IRR], [BFP], and [BX].

Low Battery Indicator

A low power condition is indicated by the **BATT** annunciator in the upper left hand side of the display. When **BATT** is lit, new batteries should be installed. Refer to your HP-71 Owner's Manual for information on replacing the batteries.

Error Messages

If you attempt a calculation using an improper parameter, such as attempting to find the square root of a negative number, an error message will appear in the display.

Keystrokes	Display	
4[CHS]	**-4**	**:F**
[√x]	**ERR:SQR(NEG)**	**:F**
[<]	**-4.0000**	**:F**

For a complete listing of error messages and their cause, refer to Appendix A, Error Messages.

To clear any error message, press [<] (or any other key), then resume normal operation.

Memory

Continuous Memory

The Continuous Memory feature of your computer maintains the following even when the computer is turned off and when you exit C71:

* All numeric data stored in C71.

* The display mode and setting.

* Flag settings.

* Trig mode (Degree or Radians).

* Calculator mode (Engineering/Business or Computer Scientist).

BASIC System Files

C71 creates two files in your HP-71 RAM:

* FORTHSYS. C71 uses a special variation of the HP-71 FORTH/Assembler ROM for some of its software. FORTHSYS is required by the FORTH system for its storage. Also the X-, Y-, Z-, T-, and LAST X registers are maintained in this file. This file is used by all special ROMFORTH applications. This file requires about 1100 bytes.

* C71RAM is also created by the FORTH system, however its use is exclusive to the C71 software. This is where all C71 related variables and storage registers are maintained. This file requires about 640 bytes.

There are 3 special classes of storage registers maintained in C71RAM:

1. User storage registers R_0 through R_9.

2. Financial registers n, i, PV, PMT, and FV.

3. Cash flow registers CF_0 through CF_{20} and the cash flow group registers N_0 through N_{20}.

Additionally, C71 uses the HP-71's internal statistical functions and internally maintained statistical arrays. The HP-71's BASIC system allocates memory for these as required.

You must have about 1800 bytes of free memory as shown by the **MEM** function in order for C71 to run.

Resetting Memory

Initially, C71 has a set of default values which it saves in C71RAM. If you wish to reset these default values to their original state, type the following:

PURGE C71RAM

while in the BASIC system of the HP-71.

Keyboard Operation

Overlays

Two overlays are provided with the C71 Calculator pac. The first is the General Functions overlay and the second is the Computer Scientist overlay. These overlays tell you where the C71 functions are available for the C71 Calculator.

Primary and Alternative Functions

Most keys on your HP-71B perform one primary and two alternate functions. With C71 two overlays have been provided, because many of the keys have been reassigned to different functions other than the standard HP-71B keyboard functions. The primary function of any key is indicated by the character on the left of the key. It is written in a vertical fashion. If no characters appear to the left of the key, the primary key is what is on the face of the key (for example: 0 through 9 and /, *, -, +). The two alternate functions are indicated by characters above the face of the key.

To select the alternate function printed in gold above the key on the left, first press the gold prefix key [f], then press the function key; for example [f][BEG].

To select the alternate function printed in blue above the key on the right, first press the blue prefix key [g], then press the function key; for example [g][END].

To select the primary function on the face of the key, press only that key, for example: [%].

Clearing Keys

Clear Stack. [CLR STK] clears automatic memory math registers: X, Y, Z, T, and LX.

Clear Registers. [CLR REG] clears the user storage registers R_0 through R_9.

Clear Financial Registers. [CLR FIN] clears the financial registers: n, i, PV, PMT, FV, CF_0 through CF_{20}, and N_0 through N_{20}.

Clear Statistics. [CLR STAT] clears the HP-71's internally maintained statistical array. C71 uses the HP-71's statistical functions and memory arrays.

Clear Prefix. If you make a mistake while keying in a prefix for a function, press the prefix key again to cancel the error.

Section 2

The Automatic Memory Stack, LX, and Data Storage

The Automatic Memory Stack and Stack Manipulation

Automatic retention and return of intermediate results is the reason C71 takes you through complex calculations so easily. The features supporting this ease of use are the automatic memory stack and the [END LINE] key.

```
The Automatic Memory Stack Registers

          T ->  0.0000
          Z ->  0.0000
          Y ->  0.0000
          X ->  0.0000        Always displayed
```

Any number keyed in and the result of executing a numeric function is placed in the displayed X-register. Executing a function or keying in a number will cause numbers already in the stack to lift, remain in the same register, or drop, depending upon the type of operation being performed. Numbers in the stack are available on a last-in, first-out basis. If the stack was loaded as shown on the left of the following illustration (as the result of previous calculations), pressing the indicated keys would result in the stack arrangement shown on the right of each illustration.

Stack Lift

```
T ->  1       2
Z ->  2       3
Y ->  3       4
X ->  4     789

Keys ->    789
```

No Stack Change

```
T ->  1       1
Z ->  2       2
Y ->  3       3
X ->  4      16
```

Keys -> [x^2]

Stack Drop

```
T ->  1       1
Z ->  2       1
Y ->  3       2
X ->  4       7

Keys ->    [+]
```

Stack Manipulation Functions

[END LINE] separates two numbers keyed in one after the other. ([END LINE] is the same as the [ENTER] key on the Hewlett-Packard series calculators.) When [END LINE] is pressed C71 lifts the stack by copying the number in the displayed X-register into the Y-register. For example, to fill the stack with the numbers 1, 2, 3, 4 (assume that the stack registers have already been loaded with the numbers shown as the result of previous calculations):

```
                     lost         lost                     lost
T ->  9               8            7            7            6
Z ->  8               7            6            6            1
Y ->  7               6            1            1            2
X ->  6               1            1            2            2

Keys            1        [END LINE]        2        [END LINE]
```

```
          lost                   lost
T ->       6           6          1          1
Z ->       1           1          2          2
Y ->       2           2          3          3
X ->       2           3          3          4

Keys            3        [END LINE]      4
```

[v] (*roll down*), [^] (*roll up*), and [x<>y] (*X exchange Y*). [v] and [^] rotate the contents of the stack registers down or up one register. No values are lost. [x<>y] exchanges the numbers in the X- and Y-registers. If the stack were loaded with the sequence 1, 2, 3, 4, the following shifts would result from pressing [v], [^], and [x<>y].

```
T ->  1        4        1        1
Z ->  2        1        2        2
Y ->  3        2        3        4
X ->  4        3        4        3

Keys       [v]      [^]      [x<>y]
```

[LX] (*LAST x*). When a numeric function is executed, a copy of the value occupying the displayed X-register before the function was executed is stored in the LX register. Pressing [f][LX] places a copy of the current contents of the LX register into the displayed X-register. For example, if the stack was loaded as shown on the left, below:

```
                          lost
T ->  0         0          0
Z ->  0         0          0
Y ->  0         0          16
X ->  4         16         4

Keys      [f][x^2]        [f][LX]

LX              4          4
```

Clearing the Stack

Pressing [f][CLR STK] clears the contents of the X-, Y-, Z-, T- and LAST X registers to zero.

C71 Functions and the Stack

When you want to key in two numbers, one after the other, you press [END LINE] between entries of the numbers. However, when you want to key in a number when the number already in the displayed X-register is the result of a previous calculation or other function (like [x<>y], [^], etc.), you do not need to use [END LINE]. Why? Executing almost any C71 function has two results:

1. The specified function is executed.
2. The automatic memory stack is enabled; that is, the stack will lift automatically when the next number is keyed in.

For example, with 4 already keyed into the X-register:

```
                        lost
T ->  0       0          0          0
Z ->  0       0          0          0
Y ->  0       0         16          0
X ->  4      16          5         21

Keys ->      [x²]        5         [+]
```

There are four functions--[END LINE], [CLRx], [Σ+], and [Σ-]--which disable the stack. They do not provide for the lifting of the stack when the next number is keyed in. Following the execution of one of these functions, keying in a new number will simply write over the currently displayed number instead of causing the stack to lift. (Although the stack lifts when [END LINE] is pressed, it will not lift when the next number is keyed in.

Two-Number Functions

An important aspect of two number functions is the positioning of the numbers in the stack. To execute an arithmetic function, the numbers should be positioned in the same way that you would write them on paper. For example, to subtract 15 from 98 you first write 98 on paper, then write 15 underneath it, like this:

```
 98
-15
```

Then you would perform the subtraction, like this:

```
 98
-15
 83
```

The numbers are positioned in the computer the same way, with the first number, the minuend, in the Y-register and the second number, the subtrahend, in the displayed X-register. When the subtraction function is executed, the 15 in the X-register is leaving the result in the X-register. Here is how the entire operation is executed (assume that the stack registers have already been loaded with the numbers shown as the result of previous calculations):

```
                    lost        lost
     T ->  4          3           2           2           2
     Z ->  3          2           1           1           2
     Y ->  2          1          98          98           1
     X ->  1         98          98          15          83

   Keys ->       98      [END LINE]     15         [-]
```

For any arithmetic function, the numbers are always positioned in their natural order first, then the function is executed and the stack drops. In the above example, we subtracted 15 from 98. The same number positioning would be used to add 15 to 98, multiply 98 by 15, and divide 98 by 15, that is:

```
                                 98
                                +15

 T ->
 Z ->                            98
 Y ->  98                       *15
 X ->  15

                                 98
                                 --
                                 15
```

Chain Calculations

It is in chain calculations that the simplicity and power of your C71's logic system becomes very apparent. Even during extremely long calculations, you still perform only one operation at a time. The automatic memory stack stores up to four intermediate results until you need them, then inserts them into the calculation. Thus, working through a problem is as natural as if you were working it out with pencil and paper.

You have already learned how to key in a pair of numbers using the **[END LINE]** key and then perform a calculation. You have seen how the stack drops as a result of executing some functions and how the stack lifts automatically when you key in a number after executing a function. To see how these features operate in a chain calculation, let's solve 3 + 6 - 4 + 2 = ? (Assume the stack cleared to zeros by pressing **[CLR STK]**.)

```
                    lost        lost
     T ->  0          0           0           0           0
     Z ->  0          0           0           0           0
     Y ->  0          0           3           3           0
     X ->  0          3           3           6           9

   Keys ->       3       [END LINE]     6          [+]

                    lost        lost
     T ->  0          0           0           0           0
     Z ->  0          0           0           0           0
     Y ->  0          9           0           5           0
     X ->  9          4           5           2           7

   Keys ->       4          [-]         2          [+]
```

As you can see, we worked through the problem one operation at a time. The stack automatically dropped after each two-number calculation. And, after each calculation, the stack automatically lifted when a new number was keyed in. Even more complicated problems are solved in the same simple manner.

Example. Instead of the arrow diagrams we've used earlier in this section, we'll use a table to follow the stack operation as we solve the expression

$$\frac{(3 + 4) \text{ X } (6 - 4)}{2}$$

T ->	0	0	0	0	0	0
Z ->	0	0	0	0	0	7
Y ->	0	3	3	0	7	6
X ->	3	3	4	7	6	6
Keys ->	<u>**3**</u>	<u>**[END LINE]**</u>	<u>**4**</u>	<u>[+]</u>	<u>**6**</u>	<u>**[END LINE]**</u>

T ->	0	0	0	0	0
Z ->	7	0	0	0	0
Y ->	6	7	0	14	0
X ->	4	2	14	2	7
Keys ->	<u>**4**</u>	<u>[-]</u>	<u>[*]</u>	<u>**2**</u>	<u>[/]</u>

LX

The C71's LAST X register, a separate data storage register, preserves the value that was last in the display before execution of a numeric function (except for the statistics functions).

T ->	d	d
Z ->	c	d
Y ->	b	c
X ->	a	a X b
Keys ->	<u>[*]</u>	
LAST X ->		a

This feature saves you from having to re-enter numbers you want to use again and can assist you in error recovery.

Example: To multiply two separate values, such as 45.575 meters and 25.331 meters by 0.175:

T ->	0.0000	0.0000	0.0000	0.0000
Z ->	0.0000	0.0000	0.0000	0.0000
Y ->	0.0000	45.5750	45.5750	0.0000
X ->	45.575	45.5750	0.175	7.9756
Keys ->	45 [.] 575	[END LINE]	[.] 175	[*]
LAST X ->				0.1750

T ->	0.0000	0.0000	0.0000
Z ->	0.0000	7.9756	0.0000
Y ->	7.9756	25.3310	7.9756
X ->	25.331	0.1750	4.4329
Keys ->	25.331	[g][LX]	[*]
LAST X ->	0.1750	0.1750	0.1750

[LX] makes it easy to recover from keystroke mistakes, such as executing the wrong function or keying in the wrong number. For example, divide 287 by 13.9 after you have mistakenly divided by 12.9:

Keystrokes	Display		
287 [END LINE]	**287.0000**	**:F**	
12.9 [/]	**22.2481**	**:F**	Oops! The wrong divisor.
[f][LX]	**12.9000**	**:F**	Retrieves from LAST X the last entry to the X-register (the incorrect divisor) before [/] was executed.
[*]	**287.0000**	**:F**	Perform the reverse of the function that produced the wrong answer.
13.9 [/]	**20.6475**	**:F**	The correct answer.

Constant Arithmetic

Because the number in the T-register remains there when the stack drops, this number can be used as a constant in arithmetic operations.

T ->	c	c
Z ->	c	c
Y ->	c	c
X ->	x	cx
Keys ->	[*]	

To insert a constant into a calculation, load the stack with the constant by keying the constant into the X-register and pressing [END LINE] three times. Use the constant by keying in your initial argument and executing your planned series of arithmetic operations. Each time the stack drops, a copy of the constant will be made available for your next calculation and a new copy of the constant is reproduced in the T-register.

Example: A bacteriologist test a certain strain of microorganisms whose population typically increases by 15% each day (a growth factor of 1.15). If he starts a sample culture of 1000, what will be the bacteria population at the end of each day for five consecutive days?

Method: Use **[END LINE]** to put the constant growth factor (1.15) in the Y-, Z-, and T-register. Then put the original population (1000) in the displayed X-register. Thereafter, you calculate the new daily population whenever you press **[*]**. To set your calculator to the same display format as is shown in the following example, press **[f][FIX] 2**.

T ->	0.00	0.00	0.00	1.15	1.15
Z ->	0.00	0.00	1.15	1.15	1.15
Y ->	0.00	1.15	1.15	1.15	1.15
X ->	1.15	1.15	1.15	1.15	1000
Keys ->	**1.15**	**[END LINE]**	**[END LINE]**	**[END LINE]**	**1000**

T ->	1.15	1.15	1.15	1.15	1.15
Z ->	1.15	1.15	1.15	1.15	1.15
Y ->	1.15	1.15	1.15	1.15	1.15
X ->	1150.00	1322.50	1520.88	1749.01	
2011.36					
Keys ->	**[*]**	**[*]**	**[*]**	**[*]**	**[*]**

When you press **[*]** the first time, you calculate 1.15 X 1000. The result (1150.00) is displayed in the X-register, the stack drops, and a new copy of the constant is generated in the T-register, that is, each time you press **[*]**:

1. A new calculation involving the X- and Y-registers takes place.	T -> c Z -> c Y -> c X -> x (C * X)
2. The result of the calculation is placed in the displayed X-register and the contents of the rest of the stack drop.	T -> c Z -> c Y -> X -> cx cx
3. A new copy of the number last in T (in this case, our constant) is generated in T.	T -> c Z -> c Y -> c X -> cx

Since a new copy of the growth factor is duplicated in the T-register each time the stack drops, you never have to re-enter it.

Press [f][FIX] 4 to return C71 to the [FIX] 4 display format.

Alternate Method: Constant arithmetic can also be performed using the LAST X register. To use this method to calculated the result of the preceding example:

1. Key in the original population (1000) and press [END LINE].

2. Key in the constant growth factor (1.15).

3. Press [*] to calculate the population at the end of the day.

4. Press [f][LX][*] to calculate the population at the end of each succeeding day.

Storage Register Operations

Storing and recalling numbers are operations involving the displayed X-register and C71's 10 data storage registers. Data storage registers are entirely separate from the stack, financial, statistical, and LX registers.

Storing Numbers

[STO] (*store*). When followed by a storage register *number* (0 through 9), copies a number from the displayed X-register into the data storage register specified by the storage register number.

Data Storage Registers

```
R0 [   ]
R1 [   ]
R2 [   ]
R3 [   ]
R4 [   ]
R5 [   ]
R6 [   ]
R7 [   ]
R8 [   ]
R9 [   ]
```

```
If...                         ... and you press [STO]0, then...

                     Data                                   Data
        Stack        Storage                   Stack        Storage

    T -> [1]         R0  [ 0 ]             T -> [1]         R0  [ 4 ]
    Z -> [2]         R1  [ 0 ]             Z -> [2]         R1  [ 0 ]
    Y -> [3]         R2  [ 0 ]             Y -> [3]         R2  [ 0 ]
    X -> [4]         R3  [   ]             X -> [4]         R3  [   ]
                          .                                      .
                          .                                      .
                          .                                      .
```

A copy of the stored number remains in the storage register until a new number is stored there or until the storage registers are cleared or Continuous Memory is reset.

Recalling Numbers

[RCL] (*recall*). When followed by a storage register *number* (0 through 9) places a copy of the number in the specified data storage register into the displayed X-register. If the stack is not disabled, executing a [RCL] operation causes the stack to lift.

```
If...                         ... and you press [RCL]2, then...

                 Data                                  Data
      Stack      Storage                  Stack        Storage
                                           1 -> lost
   T -> [1]      R0 [ 0 ]          T -> [2]          R0 [ 0 ]
   Z -> [2]      R1 [ 0 ]          Z -> [3]          R1 [ 0 ]
   Y -> [3]      R2 [ 9 ]          Y -> [4]          R2 [ 9 ]
   X -> [4]      R3 [   ]          X -> [9]          R3 [   ]
                      .                                   .
                      .                                   .
                      .                                   .
```

Storage and Recall Exercises

Execute the following operations:

Keystrokes	Display		
123	**123**	**:F**	
[STO]4	**123.0000**	**:F**	Stores 123 R_4.
678	**678**	**:F**	
[STO]7	**678.0000**	**:F**	Stores 678 in R_7.
[RCL]4	**123.0000**	**:F**	Recalls 123 from R_4.
[RCL]7	**678.0000**	**:F**	Recalls 678 from R_7.

Clearing Data Storage Registers

[CLR REG] (*clear registers*). Clears the contents of all data storage registers to zero. [CLR REG] does not affect the stack or LX registers. (To clear a single data storage register, store zero in that register.)

Storage Register Arithmetic

[STO] ([+], [-], [*], [/]) n (*storage register arithmetic*), uses the number in the displayed X-register to perform arithmetic upon the contents of a specified storage register n. The key sequence is [STO] followed by an arithmetic function key, followed in turn by the register number (0 through 9). The result of any storage register arithmetic operation is placed in the specified data storage register.

```
                                          Storage
        Stack                            Registers
T ->    [a]          and...          R0  [    ]
Z ->    [b]                          R1  [ k  ]
Y ->    [c]                          R2  [    ]
X ->    [d]                          R3  [    ]
                                           .
                                           .
                                           .
```

represents the current status of the memories, executing **[STO][*] 1** results in:

```
T ->    [a]          and...          R0  [    ]
Z ->    [b]                          R1  [ kd]
Y ->    [c]                          R2  [    ]
X ->    [d]                          R3  [    ]
                                           .
                                           .
                                           .

The same numbers                 Contents of R1 changed
in the stack                     from k to (k * d).
registers.
```

Storage Register Arithmetic Exercises

Keystrokes	Display		
18[STO]0	**18.0000**	**:F**	Stores 18 in R_0.
3[STO][/]0	**3.0000**	**:F**	Divides number in R_0 (18) by 3.
[RCL]0	**6.0000**	**:F**	Recalls copy of new number in R_0.
4[STO][*]0	**4.0000**	**:F**	Multiplies new number in R_0 (6.0000) by 4.
[RCL]0	**24.0000**	**:F**	Recalls copy of new number in R_0.
[STO][+]0	**24.0000**	**:F**	Adds 24 to number in R_0.
[RCL]0	**48.0000**	**:F**	Recalls copy of new number in R_0.
40[STO][-]0	**40.0000**	**:F**	Subtracts 40 from number in R_0.
[RCL]0	**8.0000**	**:F**	Recalls copy of new number in R_0.

Recall Register Arithmetic

[RCL] ([+], [-], [*], [/]) n (*recall register arithmetic*), uses the number in the specified storage register to perform arithmetic upon the contents of the displayed X-register. The key sequence is **[RCL]** followed by an arithmetic function key, followed in turn by the register number (0 through 9). The result of any recall register arithmetic operation is placed in the X-register overwriting the current contents of X, that is, the stack is not lifted.

```
                                        Storage
          Stack                        Registers
T ->       [a]        and...        R0  [     ]
Z ->       [b]                      R1  [  k  ]
Y ->       [c]                      R2  [     ]
X ->       [d]                      R3  [     ]
                                          .
                                          .
                                          .
```

represents the current status of the memories, executing [RCL][*] 1 results in:

```
T ->       [a]        and...        R0  [     ]
Z ->       [b]                      R1  [  k  ]
Y ->       [c]                      R2  [     ]
X ->       [dk]                     R3  [     ]
                                          .
                                          .
                                          .

The X-register                  Contents of R1 is
has changed from                unchanged.
k to (k * d).
```

Keystroke	Display		
18[STO]0	**18.0000**	**:F**	Stores 18 in R_0.
3	**3**	**:F**	Puts 3 in the X-register.
[RCL][*]0	**54.0000**	**:F**	Multiplies the number in the X-register (3) by the contents of R_0 (18.0000).

Clearing Data Storage Registers

Pressing [f][CLR REG] (*clear registers*) clears the contents of all data storage registers (R_0 through R_9) to zero. It does not affect the stack, LAST X, the financial registers, or the statics array. To clear a single data storage register, store zero in that register. Purging C71RAM will also clear the registers.

The Effects of Switching Modes

When [f][TO CS] or [f][TO EB] is pressed, the stack (X-, Y-, Z-, T-, and LAST X registers) are cleared. The contents of the data storage registers are not changed. However the numbers are stored internally in two different, incompatible formats. Therefore, any values stored in the storage registers in one format are not meaningfully converted when the calculator switches to the other format. As long as there contents are not changed while in the other format, the integrity is maintained when the calculator is back in the original mode.

Attempts to recall a value originally stored in the alternate mode will usually result in an **ERR: DATA ERROR** message. Refer to the section 10 on the Computer Scientist for more information.

Section 3
Display Control

Your C71 numeric function set enables you to perform a wide range of operations involving number alteration, math, and statistics.

Engineering/Business Display Format

When C71 is in Engineering/Business (floating point) mode (**:F** is in the display) the display can be controlled to display information in standard ([STD]), fixed ([FIX]), scientific ([SCI]), and engineering ([ENG]) formats.

[f][STD]	**123456**	**:F**
[f][FIX]4	**123456.0000**	**:F**
[f][SCI]4	**12346E5**	**:F**
[f][ENG]4	**123.46E3**	**:F**

Standard Display

[STD] (*standard*) displays the numbers in the minimum number of digits required to display the full accuracy of the number as it is stored internally. If a number is too large or too small to be displayed in 12 digits, then [SCI] (*scientific*) display mode will be used. Standard is the default setting of the calculator.

Standard display is selected by pressing [g][STD].

Fixed Decimal Display

[FIX] (*fixed decimal*) displays numbers using a fixed decimal mode without exponents. In any [FIX] display setting, C71 will automatically switch to [SCI] (*scientific*) mode to allow viewing of a displayed number that is too large or too small to be viewed in the current [FIX] mode. C71 will automatically switch back to the specified [FIX] mode when a number is displayed that can be viewed in that particular [FIX] mode display setting.

Fixed decimal display is selected or modified by pressing [f][FIX] followed by the appropriate key to specify the number of decimal places (0 to 11) to which you want the displayed rounded. For 0 through 9 use the number keys 0 through 9. For 10 and 11 use [.] and [,] respectively. However, the total number of digits displayed (excluding the exponent, if present) will never exceed 12 digits.

When you press [f][FIX] the display will show **Fixed Decimal** to remind you that you are setting fixed decimal mode and that you now need to enter the number of decimal places to display. After you press the appropriate number, the display will return to the displayed X-register in the new format. If you press any key other than 0 - 9, [.], or [,], the error message **ERR: Invalid Key** will be displayed.

Keystrokes	Display		
123.45678[END LINE]	**123.4568**	**:F**	Display is rounded to four decimal places. However, internally the number is maintained in its original value to 12 digits.
[f][FIX]6	**123.456780**	**:F**	The display is rounded upward if the first undisplayed digit is 5 or greater.
[f][FIX]0	**123.**	**:F**	
[f][FIX]4	**123.4568**	**:F**	Usual [FIX]4 display.
[f][FIX][,]	**123.456800000**	**:F**	Set to [FIX]11. 12 total digits are displayed, only 9 after the decimal point.

Scientific Notation Display

[SCI] (*scientific*) displays numbers in scientific notation mode. To select or modify a [SCI] mode, press [g][SCI] followed by the key that specifies the number of decimal places you want the display rounded to (0 to 11). For decimal places 0 through 9 use keys 0 through 9 respectively. For 10 use [.] and for 11 or [,].

When you press [g][SCI] the display will show **Scientific** to remind you that you are setting scientific mode and that you now need to enter the number of decimal places to display. After you press the appropriate number, the display will return to the displayed X-register in the new format. If you press any key other than 0 - 9, [.], or [,], the error message **ERR: Invalid Key** will be displayed.

Keystrokes	Display		
123.4567895[END LINE]	**123.4568**	**:F**	Display is rounded to 4 decimal places.
[f][SCI]2	**1.23E2**	**:F**	$1.23 * 10^2$; display rounded down.
[f][SCI]4	**1.2346E2**	**:F**	$1.2346 * 10^2$; display rounded up.

Engineering Notation Display

[ENG] (*engineering*) displays numbers in an engineering notation format that operates the same as [SCI] notation format except:

* Engineering notation shows all exponents in multiples of three.

* The number of digits specified for the display setting refers to the number of significant digits you want to appear after the leading digit.

When you press [f][ENG] the display will show **Engineering** to remind you that you are setting Engineering mode and that you now need to enter the number of significant places to display.

After you press the appropriate number, the display will return to the displayed X-register in the new format. If you press any key other than 0 - 9, [.], or [,], the error message **ERR: Invalid Key** will be displayed.

In engineering notation, the first significant digit is always present in the display. The key you press after [f][ENG] specifies the number of additional significant digits to which you want the display to be rounded. For example:

Keystrokes	**Display**		
.012345	**0.012345**	**:F**	
[f][ENG]1	**12.E-3**	**:F**	Engineering notation. Display is rounded to one significant digit after the leading digit. Power of 10 is proper multiple of three.
[f][ENG]3	**12.35E-3**	**:F**	Display is rounded to third significant digit after the leading digit.
[f][ENG]6	**12.34500E-3**	**:F**	
[f][ENG]0	**10.E-3**	**:F**	Display is rounded to first significant digit.

Notice that in [ENG] mode the decimal automatically shifts to maintain the exponent of 10 as a multiple of three, as in the case of the following example:

Keystrokes	**Display**		
[f][ENG]2	**12.3E-3**	**:F**	Display from previous example changed to [ENG]2 format.
10[*]	**123.E-3**	**:F**	Decimal shifts to maintain multiple of 3 in exponent.

Keying in Exponents

[EEX] (*enter exponent*) is used whenever an exponent is part of a number you are keying in. To use [EEX], first key in the mantissa, then press [EEX] and key in the exponent. For example, divide 95,600 by Avogadro's number ($6.0222 * 10^{26}$):

Keystrokes	**Display**		
[f][FIX]4			Reset [FIX]4 display mode.
95600[END LINE]	**95600.0000**	**:F**	
6.0222	**6.0222**	**:F**	
[EEX]	**6.0222E+**	**:F**	The cursor blinks to prompt you to key in the exponent.
26	**6.0222E+26**	**:F**	($6.0222 * 10^{26}$)
[/]	**1.5875E-22**	**:F**	kmol

To key in a number having a negative exponent of 10, first key in the number and press [EEX], then press [CHS] (change sign) to make the exponent negative, and key in the exponent. For example, key in Planck's constant (6.6262 * 10^{-34} Joule-seconds) and multiply it by 50:

Keystrokes	Display		
6.6262[EEX]	**6.6262E+**	**:F**	
[CHS]	**6.6262E-**	**:F**	
3	**6.6262E-3**	**:F**	
4	**6.6262E-34**	**:F**	
[END LINE]	**6.6262E-34**	**:F**	
50[*]	**3.3131E-32**	**:F**	Joule-seconds.

C71 and the HP-71 allows numbers containing exponents up to +499 and -499. If an exponent larger than 499 is keyed in, when that value is converted into the HP-71's internal representation, the calculator will beep and display the overflow or underflow value. Refer to "Overflow and Underflow" in section 1 for additional information.

Rounding of the Twelfth Digit

C71 holds every value to 12 digits internally, regardless of the number of digits specified in the current [FIX], [SCI], or [ENG] display setting. [STD] always displays all of the internally held significant digits. The final result of every calculation or series of calculations is rounded to the twelfth digit. For example, pi and 2/3 have nonterminating decimal representations (3.14159265359... and 0.666666666667...) Because C71 can provide only a finite approximation of such numbers (12 digits), a small error due to rounding can occur in the twelfth digit. This error can be increased through lengthy calculations, but in the majority of cases, it does not enter the practical range of significant digits for a particular application. Accurately assessing the effects of roundoff error for a given calculation requires the use of numerical analysis methods that are beyond the scope of this manual.

The HP-71 has several rounding mode options. The calculator uses the mainframes rounding settings for rounding its numbers. For further information on the HP-71's rounding capabilities and precision, refer to "Numeric Precision" in section 2, Calculating with the HP-71, in the HP-71 Owner's Manual.

Section 4
Numeric Functions

Pi

Pressing [g][pi] places a 12-digit approximation of the value of pi (3.14159265359) in the displayed X-register. If the stack is not disabled, pressing [g][pi] causes the stack to lift.

Number Alteration Functions

In addition to [CHS] C71 has four functions for altering numbers: [ABS], [IP], [FP], and [RND].

Absolute Value. Pressing [g][ABS] changes the number in the displayed X-register to the absolute value of the number.

Integer Portion. Pressing [g][IP] replaces the number in the displayed X-register with its integer portion, that is, it replaces all digits to the right of the decimal with zeros.

Fractional Portion. Pressing [f][FP] replaces the number in the displayed X-register with its decimal portion, that is, it replaces any digits to the left of the decimal with zeros.

Rounding. Pressing [g][RND] rounds the internally held 12-digit mantissa of any displayed value to the number of digits specified by the current [FIX], [SCI], or [ENG] display setting.

To Calculate	Keystroke Example	Display
Absolute Value	12345 [CHS]	**-12345 :F**
	[g][ABS]	**12345.0000 :F**
Integer Portion	123.4567	**123.4567 :F**
	[g][IP]	**123.0000 :F**
Fractional Portion	123.4567	**123.4567 :F**
	[f][FP]	**0.4567 :F**
Round (Assumes	1.23456789	**1.23456789 :F**
[FIX]4.)	[g][RND]	**1.2346 :F**
Inspect rounding	[f][FIX]8	**1.23460000 :F**
Reset to [FIX] 4	[f][FIX]4	**1.2346 :F**

One-Number Functions

C71's one-number math functions have the following characteristics:

* Use the number in the displayed X-register as the argument for the function.
* Replace the number in the displayed X-register with the result of executing the function.
* Do not affect numbers in the Y-, Z-, and T-registers.

General Functions

Reciprocal. Pressing [1/x] calculates the reciprocal of the number in the displayed X-register, that is, it divides 1 by the number in the displayed X-register.

Factorial. Pressing [g][x!] calculates the factorial value as follows:

When executed with a positive integer n (0 <= n <= 253) in the displayed X-register, [x!] calculates the factorial of n, that is, it calculates the product of the integers from 1 to n.

Square Root. Pressing [f][√x] calculates the square root of the number in the displayed X-register.

Squaring. Pressing [X^2] calculates the square of the number in the displayed X-register.

To Calculate	Keystroke Example	Display	
Reciprocal	25	25	:F
	[1/x]	0.0400	:F
Factorial	8	8	:F
	[x!]	40320.0000	:F
Square Root	3.9	3.9	:F
	[x]	1.9748	:F
Square	12.3	12.3	:F
	[f][X^2]	151.2900	:F

Trigonometric Operations

The six basic trigonometric function operate in the trigonometric mode you select.

Trigonometric Modes. Selecting a specific trig mode does not convert any number already in the computer to that mode. Selecting a specific trig mode is simply telling the computer what unit of measure (degree or radian) to use when executing a trig function.

Pressing [f][DEG] selects the Degree trig mode. No annunciator appears in the display.

Press [g][RAD] selects the Radian trig mode. While [RAD] mode is set, the RAD annunciator appears in the display.

The computer is always set to one of the two trig modes. Continuous memory maintains the last trig mode selected, even when you exit C71. If a power loss occurs, or if you reset Continuous Memory, the computer will automatically reset to [DEG] mode.

Trigonometric Functions.

Pressing	Calculates
[SIN]	sine
[f][ASIN]	arc sine
[COS]	cosine
[f][ACOS]	arc cosine
[TAN]	tangent
[f][ATAN]	arc tangent

To use any of the trig functions, ensure that the computer is set to the desired trig mode (**[DEG]**, or **[RAD]**) then execute the desired trig function.

To Calculate (Examples assume [DEG] trig mode and [FIX] 4.)	**Keystroke Example**	**Display**	
All Trig functions, for example:			
Sine	33.5	**33.5**	**:F**
	[SIN]	**0.5519**	**:F**
Arc Sine	.7982	**0.7982**	**:F**
	[f][ASIN]	**52.9586**	**:F**

Time and Angle Conversions

Numbers representing time or angles are interpreted in a decimal or minutes-seconds format, depending upon the conversion being executed:

Hours.Decimal Hours (H.h)	Hours.Minutes Seconds Decimal-seconds (H.MMSSs)
or	or
Degrees.Decimal Degrees (D.d)	Degrees.Minutes Seconds Decimal-seconds (D.MMSSs)

Hours (or Degrees), Minutes, Seconds Conversion. Pressing **[f][->H.MS]** converts the number in the displayed X-register from a decimal hours (or decimal degrees) format to an hours (or degrees), minutes, seconds, decimal seconds format.

```
H.h -> H.MMSSs
     or
D.d -> D.MMSSs
```

Decimal Hours (or Degrees) Conversion. Pressing **[g][->H]** converts the number in the displayed X- register from an hours (or degrees), minutes, seconds, decimal seconds format to a decimal hour (or degrees) format.

H.MMSSs -> H.h
or
D.MMSSs -> D.d

Hours (or Degrees), Minutes, Seconds Arithmetic. Pressing **[g][H.MS+]** will add an H.MMSSs format number in the Y-register to an H.MMSSs format number in the X-register. Likewise, pressing **[g][H.MS-]** will subtract an H.MMSSs number in the X-register from an H.MMSSs format number in the Y-register.

Degrees/Radians Conversions

The **[->DEG]** and **[->RAD]** functions are used to convert angles between decimal degrees and radians (D.d -> R.r and R.r -> D.d).

Degrees to Radians Conversions. Pressing **[g][->RAD]** converts the number in the displayed X- register from a decimal degree value to its radian equivalent.

Radians to Degrees Conversion. Pressing **[f][->DEG]** converts the number in the displayed X-register from a radian value to its decimal degree equivalent.

To Convert	Example Keystrokes	Display	
Decimal hours (H.h) or degrees (D.d) to H.MMSSs or D.MMSSs format.	17.553	**17.553**	**:F**
	[f][->H.MS]	**17.3311**	**:F**
To view decimal seconds while in [FIX]4 setting, press:	[f][STD]	**17.33108**	**:F**
	[f][FIX]4	**17.3311**	**:F**
H.MMSSs or D.MMSSs to decimal hours (H.h) or degrees (D.d) format.	12.3045	**12.3045**	**:F**
	[g][->H]	**12.5125**	**:F**
H.MMSSs addition	12.5948[END LINE]	**12.5948**	**:F**
	1.0030	**1.0030**	**:F**
	[g][H.MS+]	**14.0018**	**:F**
Degrees to Radians	40.5	**40.5**	**:F**
	[g][->RAD]	**0.7069**	**:F**
Radians to Degrees	1.1746	**1.1746**	**:F**
	[f][->DEG]	**67.2996**	**:F**

Logarithmic Functions

Natural Logarithm. Pressing [f][LN] calculates the natural logarithm of the number in the displayed X-register, that is, the logarithm to the base e (2.71828182846) of the number in the displayed X-register.

Natural Antilogarithm. Pressing [e^x] calculates the natural antilogarithm of the number in the displayed X-register, that is, it raises e (2.71828182846) to the power of the value in the X-register.

Natural Logarithm X Plus 1. Pressing [f][LOGP1] calculates the quantity ln(1 + x), which is useful for values in the X-register that are close to one.

Natural Antilog Minus 1. Pressing [EXPM1] calculates the quantity e^x - 1, which is useful for values in X-register that are close to zero.

Common Logarithm. Pressing [f][LOG] calculates the common logarithm of the number in the displayed X-register, that is, the logarithm to the base 10.

Common Antilogarithm. Pressing [10^x] calculates the common antilogarithm of the number in the displayed X-register, that is, it raise 10 to the power of that number.

To Calculate	Example Keystrokes	Display
Natural Log	45	**45 :F**
	[f][LN]	**3.8067 :F**
Natural Antilog	3.4012	**3.4012 :F**
	[e^x]	**30.0001 :F**
Natural Log Plus 1	2.3456E-10	**2.3456E-10 :F**
	[f][LOGP1]	**2.3455E-10 :F**
Natural Antilog Minus 1	0.0001	**0.0001 :F**
	[g][SCI]4	**1.0000E-4 :F**
	[EXPM1]	**1.0001E-4 :F**
	[f][FIX]4	**0.0001 :F**
Common Log	12.4578	**12.4578 :F**
	[f][LOG]	**1.0954 :F**
Common Antilog	3.1354	**3.1354 :F**
	[10^x]	**1365.8405 :F**

Hyperbolic Functions

Pressing	Calculates
[HYP][SIN]	Hyperbolic sine (sinh)
[f][AHYP][SIN]	Inverse hyperbolic sine (asinh or $\sinh^{-1}$)
[HYP][COS]	Hyperbolic cosine (cosh)
[f][AHYP][COS]	Inverse hyperbolic cosine (acosh or $\cosh^{-1}$)
[HYP][TAN]	Hyperbolic tangent (tanh)
[f][AHYP][TAN]	Inverse hyperbolic tangent (atanh or $\tanh^{-1}$)

To Calculate	Example Keystrokes	Display	
All hyperbolic functions; for example:			
Hyperbolic sine	2.53	2.53	:F
	[HYP][SIN]	6.2369	:F
Inverse hyperbolic sine	1.95	1.95	:F
	[f][AHYP][SIN]	1.4210	:F

Two-Number Functions

The C71 two-number functions use the values in the displayed X-register and the Y-register to calculate a result. To use any of these functions, key in the Y-register value first, press [END LINE] to lift the value into the Y-register, key in the displayed X-register value, then execute the function.

Exponential

Pressing [Y^X] raises the number in the Y-register to the power of the number in the X-register.

To Calculate	Example Keystrokes	Display	
Exponential	2[END LINE]	2.0000	:F
	3	3	:F
	[Y^X]	8.0000	:F

Polar-Rectangular Coordinate Conversions

Two functions ([->P], [->R]) are provided for polar- rectangular coordinate conversions.

```
                  x
              |------/
              |  r / |
              |  /   | y
              |/  Θ  |
       ----------------------
              |
              |
              |
```

Angle Θ is assumed to be in decimal degrees or radians depending upon which trigonometric mode ([DEG], or [RAD]) the computer is set to. Angle Θ is measured as shown in the illustration above. The answer returned for Θ is between 180 degrees and -180 degrees.

Polar Conversion. Pressing [g][->P] (polar) converts values in the X- and Y- registers representing rectangular coordinates (x, y) to polar coordinates (magnitude r, angle Θ).

Rectangular Conversion. Pressing [f][->R] (rectangular) converts values in the X- and Y- registers representing polar coordinates (magnitude r, and angle Θ), to rectangular coordinates (x,y).

To Calculate	Example Keystrokes	Display	
Rectangular coordinates to polar:			
y	5[END LINE]	**5.00**	**:F**
x	10	**10**	**:F**
r	[g][->P]	**11.1803**	**:F**
Θ	[x<>y]	**26.5651**	**:F**
Polar coordinates to rectangular:			
Θ	30[END LINE]	**30.0000**	**:F**
r	12	**12**	**:F**
x	[f][->R]	**10.3923**	**:F**
y	[x<>y]	**6.0000**	**:F**

Section 5
Percentage and Calendar Functions

Percentage Functions

C71 includes three keys for solving percentage problems: [%], [Δ%], [%T]. You don't need to convert percentages to their decimal equivalents; this is done automatically when you press any of these keys. Thus 4% need not be changed to .04; you key it in the way you see and say it: 4[%].

Percentages

To find the amount corresponding to a percentage of a number:

1. Key in the base number.
2. Press [END LINE].
3. Key in the percentage.
4. Press [%].

For example, to find 14% of $300:

Keystrokes	Display		
These examples presume [FIX] 2 display format.			
300	**300**	:F	Keys in the base number.
[END LINE]	**300.00**	:F	Pressing [END LINE] separates the next number entered from the first number, just as when an ordinary arithmetic calculation is performed.
14	**14**	:F	Keys in the percentage.
[%]	**42.00**	:F	Calculates the amount.

If the base number is already in the display as a result of a previous calculation, you should not press [END LINE] before keying in the percentage--just as in a chain arithmetic calculation.

Net Amount

A net amount--that is, the base amount plus or minus the percentage amount--can be calculated easily with the C71 software, since the computer holds the base amount inside after you calculate a percentage amount. To calculate a net amount, simply calculate the percentage amount, then press [+] or [-].

Example: You're buying a new car that lists for $13,250. The dealer offers you a discount of 8%, and the sales tax is 6%. Find the amount the dealer is charging you, then find the total cost to you, including tax.

Keystrokes	Display		
13250 [END LINE]	**13250.00**	:F	Keys in the base amount and separates it from the percentage.
8[%]	**1060.00**	:F	Amount of discount.
[-]	**12190.00**	:F	Base amount less discount.
6[%]	**731.40**	:F	Amount of tax (on $12,190).
[+]	**12921.40**	:F	Total cost: base amount less discount plus tax.

Percent Difference

To find the percent difference between two numbers:

1. Key in the base number.
2. Press [END LINE] to separate the other number from the base number.
3. Key in the other number.
4. Press [Δ%].

If the other number is greater than the base number, the percent difference will be positive. If the other number is less than the base number, the percent difference will be negative. Therefore, a positive answer indicates an increase, while a negative answer indicates a decrease.

If you are calculating a percent difference over time, the base number is typically the amount occurring first.

Example: Yesterday your stock fell from 58 1/2 to 53 1/4 per share. What is the percentage change?

Keystrokes	Display		
58.5 [END LINE]	**58.50**	:F	Keys in the base number and separates it from the other number.
53.25	**53.25**	:F	Keys in the other number.
[Δ%]	**-8.97**	:F	Nearly a 9% decrease.

The [Δ%] key can be used for calculations of the percent difference between a wholesale cost and a retail cost. If the base number entered is the wholesale cost, the percent difference is called the *markup*; if the base number entered is the retail cost, the percent difference is called the *margin*.

Percent of Total

To calculate what percentage one number is of another:

1. Calculate the total amount by adding the individual amounts, just as in a chain arithmetic calculation.
2. Key in the number whose percentage equivalent you wish to find.
3. Press [%T].

Example: Last month, your company posted sales of $3.92 million in the U.S., $2.36 million in Europe, and $1.67 million in the rest of the world. What percentage of the total sales occurred in Europe?

Keystrokes	Display		
3.92 [END LINE]	**3.92**	**:F**	Keys in the first number and separates it from the second.
2.36 [+]	**6.28**	**:F**	Adds the second number.
1.67 [+]	**7.95**	**:F**	Adds the third number to get the total.
2.36	**2.36**	**:F**	Keys in 2.36 to find what percentage it is of the number in the display.
[%T]	**29.69**	**:F**	Europe had nearly 30% of the total sales.

The C71 software retains the total amount inside after a percent of total is calculated. Therefore, to calculate what percentage another amount is of the total:

1. Clear the display by pressing [CLRx].
2. Key in the amount.
3. Press [%T] again.

For example, to calculate what percent of the total sales in the preceding example occurred in the U.S. and what percent occurred in the rest of the world:

Keystrokes	Display		
[CLRx] 3.92 [%T]	**49.31**	**:F**	The U.S. had about 49% of the total sales.
[CLRx] 1.67 [%T]	**21.01**	**:F**	The rest of the world had about 21% of the total sales.

To find what percentage a number is of a total, when you already know the total number:

1. Key in the total number.
2. Press [END LINE] to separate the other number from the total number.
3. Key in the number whose percentage equivalent you wish to find.
4. Press [%T].

For example, if you already knew in the preceding example that the total sales were $7.95 million and you wanted to find what percentage of the total occurred in Europe:

Keystrokes	Display		
7.95 [END LINE]	**7.95**	**:F**	Keys in the total amount and separates it from the next number.
2.36	**2.36**	**:F**	Keys in 2.36 to find what percentage it is of the number in the display.
[%T]	**29.69**	**:F**	Europe had nearly 30% of the total sales.

Calendar Functions

The calendar functions provided by C71--[DATE] and [ΔDYS]--can handle dates from January 1, 0000 to December 31, 9999. However dates before October 15, 1582--before January 1, 1752 for English speaking countries--do not relate directly to our Gregorian calendar. The HP-71 calendar functions are used to do these calculations and further information may be obtained by referring to the HP-71 Owner's Manual section 5.

Date Format

For each of the calendar functions the computer uses one of the two date formats. The date format is used to interpret dates when they are keyed into the computer as well as for displaying dates.

Month-Day-Year. To set the date format to month-day-year, press [g][M.DY] to key in a date with this format in effect:

1. Key in the one or two digits of the month.
2. Press the decimal point key ([.]).
3. Key in the *two* digits of the day.
4. Key in the four digits of the year.

Dates are displayed in the same format. For example, to key in April 7, 1986:

Keystrokes	Display	
4.071986	**4.071986**	**:F**

Day-Month-Year. To set the date format to day-month-year, press [f][D.MY]. To key in a date with this format in effect:

1. Key in the one or two digits of the day.
2. Press the decimal point key ([.]).
3. Key in the *two* digits of the month.
4. Key in the four digits of the year.

For example, to key in 7 April 1986:

Keystrokes	Display	
7.041986	**7.041986**	**:F**

When the date format is set to day-month-year, the D.MY status indicator (flag 1) is lit. If flag 1 is not lit, the date format is set to month-day-year.

The date format remains set to what you last specified until you change it; it is not reset each time the computer is turned on. However, if Continuous Memory is reset or C71RAM is purged, the date format is set to month-day-year.

Future or Past Dates

To determine the date and day that is a given number of days from a given date:

1. Key in the given date and press [END LINE].
2. Key in the number of days.
3. If the other date is in the past, press [CHS].
4. Press [f][DATE].

The answer calculated by the [DATE] function is displayed in a special format. The numbers of the month, day, and year (or day, month, and year) are separated by a slash (/) and an abbreviation for the day of the week is shown to the right: Mon for Monday through Sun for Sunday.

Example: If you purchased a 120-day option on a piece of land on 14 May 1982, what would be the expiration date? Assume that you normally express dates in the day-month-year format.

Keystrokes	Display		
[f][D.MY]	**7.04**	**:F**	Sets date format to day-month- year. (Display shown assumes date remains from preceding example. The full date is not now displayed because the display format is set to show only two decimal places.)
14.051981 [END LINE]	**14.05**	**:F**	Keys in date and separates it from number of days to be entered.
120[f][DATE]	**11/9/1981 FRI**	**:F**	The expiration date is 11 September 1981, a Friday.
[<]	**11.09**	**:F**	Press any key to clear the special display format.

Number of Days Between Dates

To calculate the number of days between two given dates:

1. Key in the earlier date and press [END LINE].
2. Key in the later date and press [g][ΔDYS].

The answer shown in the display is the actual number of days between the two dates, including leap days (the extra days occurring in leap years), if any. In addition, the software also calculates the number of days between the two dates on the basis of a 30-day month. This answer is held inside the computer; to display it, press [x<>y]. Pressing [x<>y] again will return the original answer to the display.

Example: Simple interest calculations can be done using either the actual number of days or the number of days counted on the basis of a 30-day month. What would be the number of days counted each way, to be used in calculating the simple interest accruing from June 3, 1983 to October 15, 1984? Assume that you normally express dates in the month-day-year-format.

Keystrokes	Display		
[g][M.DY]	**11.09**	**:F**	Sets date format to month-day-year. (Display shown assumes date remains from preceding example.)
6.031983 [END LINE]	**6.03**	**:F**	Keys in earlier date and separates it from the later date.
10.151984 [g][ΔDYS]	**500.00**	**:F**	Keys in later date. Display shows actual number of days.
[x<>y]	**492.00**	**:F**	Number of days counted on the basis of a 30-day month.

Section 6
Basic Financial Functions

The Financial Registers

In addition to the data storage register discussed in section 2, the C71 software has five special registers in which numbers are stored for financial calculations. These registers are designated n, i, PV, PMT, and FV. The first five keys on the top left row of the computer are used to store a number from the display into the corresponding register, to calculate the corresponding financial value and store the result into the corresponding register, or to display the number stored in the corresponding register.

> **Note:** Which operation is performed when one of these keys is pressed depends upon the last preceding operation performed: If a number was just stored into a financial register (using [n], [i], [PV], [PMT], [FV], [12*], or [12/]), pressing one of these five keys calculates the corresponding value and stores it into the corresponding register; otherwise, pressing one of these five keys merely stores the number from the display into the corresponding register.

Storing Numbers Into the Financial Registers

To store a number into a financial register, key the number into the display, then press the corresponding key (n, i, PV, PMT, or FV).

Displaying Numbers in the Financial Registers

To display a number stored in a financial register, press [RCL] followed by the desired register key.

> **Note:** It is a good practice to press the corresponding key *twice* after [RCL], since often you may want to calculate a financial value right after displaying another financial value. As indicated in the preceding note, if you wanted to display FV and then calculate PV, for example, you should press [RCL][FV][FV][PV]. If you did not press [FV] the second time, pressing [PV] would store FV in the PV register rather than calculating PV, and to calculate PV you would have to press [PV] again.

Clearing the Financial Registers

Every financial function uses numbers stored in several of the financial registers. Before beginning a new financial calculation, it is good practice to clear all of the financial registers by pressing [f][CLR FIN]. [f][CLR FIN] clears only the financial registers, and does not clear any storage registers. Frequently, however, you may want to repeat a calculation after changing a number in only one of the financial registers. To do so, do not press [f][CLR FIN]; instead, simply store the new number in the register. The numbers in the other financial registers remain unchanged.

Simple Interest Calculations

The C71 software simultaneously calculates simple interest on both a 360-day basis and a 365-day basis. You can display either one, as described below. Furthermore, with the accrued interest in the display, you can calculate the total amount (principal plus accrued interest) by pressing [+].

1. Key in or calculate the number of days, then press [n].
2. Key in the annual interest rate, then press [i].
3. Key in the principal amount, then press [CHS][PV].
4. Press [f][INT] to calculate and display the interest accrued on a 360-day basis.
5. If you want to display the interest accrued on a 365-day basis, press [v][x<>y].
6. Press [+] to calculate the total of the principal and the accrued interest now in the display.

The quantities n, i, and PV can be entered in any order.

Example 1: Your good friend needs a loan to start his latest enterprise and has requested that you lend him $450 for 60 days. You lend him the money at 7% simple interest, to be calculated on a 360-day basis. What is the amount of accrued interest he will owe you in 60 days, and what is the total amount owed?

Keystrokes	**Display**		
60[n]	**60.00**	**:F**	Stores the number of days.
7[i]	**7.00**	**:F**	Stores the annual interest rate.
450[CHS][PV]	**-450.00**	**:F**	Stores the principal.
[f][INT]	**5.25**	**:F**	Accrued interest, 360-day basis.
[+]	**455.25**	**:F**	Total amount; principal plus accrued interest.

Example 2: Your friend agrees to 7% interest on the loan from the preceding example, but asks that you compute it on a 365-day basis rather than a 360-day basis. What is the amount of accrued interest he will owe you in 60 days, and what is the total amount owed?

Keystrokes	**Display**		
60[n]	**60.00**	**:F**	If you have not altered the
7[i]	**7.00**	**:F**	numbers in the n, i, and PV
450[CHS][PV]	**-450.00**	**:F**	registers since the preceding example, you may skip these first 3 steps.
[f][INT][v][x<>y]	**5.18**	**:F**	Accrued interest 365-day basis.
[+]	**455.18**	**:F**	Total amount: principal plus accrued interest.

Financial Calculations and the Cash Flow Diagram

The concepts and examples presented in this section are representative of a wide range of financial calculations. If your specific problem does not appear to be illustrated in the pages that follow, don't assume that the computer is not capable of solving it. Every financial calculation involves certain basic elements; but the terminology used to refer to these elements typically differs among the various segments of the business and financial communities. All you need to do is identify the basic elements in your problem, and then structure the problem so that it will be readily apparent what quantities you need to tell the computer and what quantity you want to solve for.

An invaluable aid for using your computer in a financial calculation is the *cash flow diagram.* This is simply a pictorial representation of the timing and direction of financial transactions, labeled in terms that correspond to keys on the computer.

The diagram begins with a horizontal line, called a *time line.* It represents the duration of a financial problem, and is divided into compounding periods. For example, a financial problem that transpires over 6 months with monthly compounding would be diagramed like this:

```
|-----|-----|-----|-----|-----|-----|
   1     2     3     4     5     6
```

The exchange of money in a problem is depicted by vertical arrows. Money you receive is represented by an arrow pointing up from the point in the time line when the transaction occurs; money you pay out is represented by an arrow pointing down.

```
      ^
      |
      |              Money paid out
-|---------|-        -|---------|-
Money Received                |
                              |
                              v
```

Suppose you deposited (paid out) $1,000 into an account that pays 6% annual interest and is compounded monthly, and you subsequently deposited an additional $50 at the end of each month for the next 2 years. The cash flow diagram describing the problem would look like this:

```
                                                  FV ?
                          i = 6/12            ^
                                              |
                                              |
   |-----|-----|---    ...    ---|-----|-----|
   |     |     |                 |     |     |
   |     v     v                 v     v     v
   v    PMT                             n = 2 * 12
  PV     50
1,000
```

The arrow pointing up at the right of the diagram indicates that money is received at the end of the transaction. Every completed cash flow diagram must include at least one cash flow in each direction. Note that cash flows corresponding to the accrual of interest are not represented by arrows in the cash flow diagram.

The quantities in the problem that correspond to the first five keys on the top left row of the keyboard are now readily apparent from the cash flow diagram:

* *n* is the number of compounding periods. This quantity can be expressed in years, months, days, or any other time unit, as long as the interest rate is expressed in terms of the same basic compounding period. In the problem illustrated in the cash flow diagram above, n = 2 * 12.

* *i* is the interest rate *per compounding period.* The interest rate shown in the cash flow diagram and entered into the computer is determined by dividing the annual interest rate by the number of compounding periods. In the problem illustrated above, i = 6% / 12.

* *PV*--the *present value*--is the initial cash flow or the present value of a series of future cash flows. In the problem illustrated above, PV is the $1,000 initial deposit.

* *PMT* is the periodic *payment.* In the problem illustrated above, PMT is the $50 deposited each month. When all payments are equal, they are referred to as *annuities*. (Problems involving equal payment are descried in this section under Compound Interest Calculation; problems involving unequal payment can be handled as described in section 7 under Discounted Cash Flow Analysis.)

* FV--the *future value*--is the final cash flow or the compounded value of a series of prior cash flows. In the particular problem illustrated above, FV is unknown (but can be calculated).

Solving the problem is now basically a matter of keying in the quantities identified in the cash flow diagram using the corresponding keys, and then calculating the unknown quantity by pressing the corresponding key. In the particular problem illustrated in the cash flow diagram above, FV is unknown quantity; but in other problems, as we shall see later, n,i, PV or PMT could be the unknown quantity. Likewise, in the particular problem illustrated above there are four known quantities that must be entered into the computer before solving for the unknown quantity; but in other problems only three quantities may be known--which must always include n or i.

The Cash Flow Sign Convention

When entering the PV, PMT, and FV cash flows, the quantities must be keyed into the computer with the proper sign, + (plus) or - (minus) in accordance with ...

> **The Cash Flow Sign Convention:** Money received (arrow pointing up) is entered or displayed as a positive value (+). Money paid out (arrow pointing down) is entered or displayed as a negative value (-).

The Payment Mode

One more bit of information must be specified before you can solve a problem involving periodic payments. Such payments can be made either at the beginning of a compounding period (payments in advance, or annuities due) or at the end of the period (payments in arrears, or ordinary annuities). Calculations involving payments in advance yield different results than calculations involving payments in arrears. Illustrated below are portions of cash flow diagrams in arrears (End). In the problem illustrated in the cash flow diagram above, payments are made in arrears.

```
              Begin                                   End
     |-----|-----|-----|                  |-----|-----|-----|
     |  1  |  2  |  3                        1  |  2  |  3  |
     v     v     v                              v     v     v
```

Regardless of whether payments are made in advance or in arrears, the number of payments must be the same as the number of compounding periods.

To specify the payment mode:

* Press [f][BEG] if payments are made at the beginning of the compounding periods.

* Press [g][END] if payments are made at the end of the compounding periods.

The **BEGIN** status indicator (flag 0) is lit when the payment mode is set to **BEGIN**. If flag 0 is not lit, the payment mode is set to End.

The payment mode remains set to what you last specified until you change it; it is not reset each time the calculator is turned on. However, if Continuous Memory is reset or C71RAM is purged, the payment mode will be set to End.

Generalized Cash Flow Diagrams

Examples of various kinds of financial calculations, together with the applicable cash flow diagrams, appear under Compound Interest Calculations later in this section. If your particular problem does not match any of those shown, you can solve it nevertheless by first drawing a cash flow diagram, then keying the quantities identified in the diagram into the corresponding registers. Remember always to observe the sign convention when keying in PV, PMT, and FV.

The terminology used for describing financial problems varies among the different segments of the business and financial communities. Nevertheless, most problems involving compound interest can be solved by drawing a cash flow diagram in one of the following basic forms. Listed below each form are some of the problems to which the diagram applies.

```
                         ^                                ^
                         |                                |
                         |                                |
|-----|-----|-----|-----|         |-----|-----|-----|-----|
|                                 |     |     |     |
v                                 v     v     v     v
    Compound Growth                  Savings Plan
    Savings Account                  Pension Fund
      Appreciation                    Annuity Due
```

```
^
|
|
|-----|-----|-----|-----|
      |     |     |     |
      v     v     v     v

            Mortgage
Direct Reduction (Installment) Loan
          Amortization
        Ordinary Annuity
```

```
^
|
|
|-----|-----|-----|-----|
      |     |     |     |
      v     v     v     v
                        v
   Mortgage with Balloon
       Amortization
     Ordinary Annuity
```

```
^
|
|
|-----|-----|-----|-----|
|     |     |     |
v     v     v     v
         Lease
      Amortization
      Annuity Due
```

```
   ^
   |
   |
   |-----|-----|-----|-----|
   |     |     |     |     |
   v     v     v     v     |
                           v
Lease With Buyback (Residual)
        Amortization
        Annuity Due
```

Compound Interest Calculations

Specifying the Number of Compounding Periods and the Periodic Interest Rate

Interest rates are usually quoted at the *annual rate* (also called the *nominal rate*): that is, the interest rate per year. However, in compound interest problems, the interest rate entered into i must always be expressed in terms of the basic compounding period, which may be years, months, days, or any other time unit. For example, if a problem involves 6% annual interest compounded quarterly for 5 years, n--the number of quarters--would be 5 * 4 = 20 and i--the interest rate per quarter--would be 6%/4 = 1.5%. If the interest were instead compounded monthly, n would be 5 * 12 = 60 and i would be 6%/12 = 0.5%.

If you use the computer to multiply the number of years by the number of compounding periods per year, pressing [n] then stores the results into *n*. The same is true for *i*.

If interest is compounded monthly, you can use a shortcut provided on the computer to calculate and store *n* and *i*:

* To calculate and store *n*, key the number of years into the display, then press [g][12*].
* To calculate and store *i*, key the annual rate into the display, then press [g][12/].

Note that these keys not only multiply or divide the displayed number by 12; they also automatically store the result in the corresponding register, so you need not press the [n] or [i] key next.

Calculating the Number of Payments or Compounding Periods

1. Press [f][CLR FIN] to clear financial registers.
2. Enter the periodic interest rate, using [i] or [12/].
3. Enter at least two of the following values:
   * Present value, using [PV].
   * Payment amount, using [PMT].
   * Future value, using [FV].

   Note: Remember to observe the cash flow sign convention.
4. If a PMT was entered, press [f][BEG] or [g][END] to set the payment mode.
5. Press [n] to calculate the number of payments or periods.

If the answer calculated is not an integer (that is, there are nonzero digits to the right of the decimal point), it means that it will take at least that number of periods. The part after the decimal point *does not* mean that it takes that fraction of a final payment. For example if payments of $100 are made and *n* is calculated to 5.45, the last payment *is not* .45 X $100. Refer to the following examples show how to properly calculate the final payment.

Example 1: You are planning to build a log cabin on your vacation property. Your rich uncle offers you a $35,000 loan at 10.5% interest. If you make $325 payments at the end of each month, how many payments will be required to pay off the loan, and how many years will this take?

```
35,000
  PV
   ^
   |             i = 10.5/12
   |
   |-----|-----|--  ...  --|-----|-----|
         |     |           |     |     |
         v     v           v     v     v
        PMT                           n = ?
       -325
```

Keystrokes	Display		
[f][CLR FIN]			
10.5[g][12/]	**.88**	**:F**	Calculates and stores i.
35000[PV]	**35000.00**	**:F**	Stores PV.
325[CHS][PMT]	**-325.00**	**:F**	Stores PMT (with minus sign for cash paid out).
[g][END]	**-325.00**	**:F**	Sets the payment mode to END.
[n]	**327.44**	**:F**	Number of full payments required is 327.
12[/]	**27.29**	**:F**	Twenty-seven years and three months.

While 328 payments will be required to pay off the loan--only 327 full payments of $325 will be required, the next and final payment being less than $325. You can calculate the final, fractional, 328th payment as follows:

Keystrokes	Display		
328[n]	**328.00**	**:F**	Stores total number of payments.
[FV]	**181.89**	**:F**	Calculates FV--which equals the over payment if 328 full payments were made.
[RCL][PMT]	**-325.00**	**:F**	Recalls payment amount.
[+]	**-143.11**	**:F**	Final, fractional payment.

Alternatively, you could make the fractional payment together with the 327th payment. (Doing so will result in a somewhat smaller total of all payments, since you will not have to pay interest during the 328th payment period.) You can calculate this final, larger, 327th payment (essentially a balloon payment) as follows:

Keystrokes	Display		
327[n]	**327.00**	**:F**	Stores number of full payments.
[FV]	**-141.87**	**:F**	Calculates FV--which is the balance remaining after 327 full payments.
[RCL][PMT]	**-325.00**	**:F**	Recalls payment amount.
[+]	**-466.87**	**:F**	Final balloon payment.

Example 2: You are opening a savings account today (the middle of the month) with $775 deposit. The account pays 6 1/4% interest compounded semimonthly. If you make semimonthly deposits of $50 beginning next month, how long will it take for your account to reach $4000?

```
                                                4,000
                                              ^   FV
                          i = 6.25/24         |
                                              |
        |-----|-----|--  ...  --|-----|-----|
        |     |     |           |     |     |
        |     v     v           v     v     v
        v    PMT                            N = ?
       PV    -50
      -775
```

Keystrokes	Display		
[f][CLR FIN]			
6.25[END LINE]24[/][i]	**0.26**	:F	Calculates and stores i.
775[CHS][PV]	**-775.00**	:F	Stores PV (with minus sign for cash paid out).
50[CHS][PMT]	**-50.00**	:F	Stores PMT (with minus sign for cash paid out).
4000[FV]	**4000.00**	:F	Stores FV.
[g][END]	**4000.00**	:F	Sets the payment mode to End.
[n]	**57.55**	:F	Number of full semimonthly deposits is 57.
2[/]	**28.77**	:F	Number of months is 28 3/4.

As in Example 1, it is likely that only 57 full deposits will be required, the next and final deposit being less than $50. You can calculate the final, fractional, 58th deposit as in Example 1, except that for this example you must subtract the original FV. (In Example 1, the original FV was zero.) The procedure is as follows:

Keystrokes	Display		
58[n]	**58.00**	:F	58 payments
[FV]	**4027.27**	:F	Calculates FV--which equals the balance in the account if 58 full deposits were made.
[RCL][PMT]	**-50.00**	:F	Recalls amount of deposits.
[+]	**3977.27**	:F	Calculates the balance in the account if 57 full deposits were made and interest accrued during the 58th month.
4000[-]	**-22.73**	:F	Calculates final fractional, 58th deposit required to reach $4000.

(You might think that we could calculate the balance in the account after 57 full deposits were made simply by storing that number in n and then calculating FV. However, this balance would not include the interest accrued during the 58th month.)

Calculating the Periodic and Annual Interest Rates

1. Press [f][CLR FIN] to clear the financial registers.
2. Enter the number of payments or periods, using [n] or [12*].
3. Enter at least two of the following values:
   * Present value, using [PV].
   * Payment amount, using [PMT].
   * Future value, using [FV].

   Note: Remember to observe the cash flow sign convention.
4. If a PMT was entered, press [f][BEG] or [g][END] to set the payment mode.
5. Press [i] to calculate the periodic interest rate.
6. To calculate the annual interest rate, key in the number of periods per year, then press [*].

Example: What annual interest rate must be obtained to accumulate $10,000 in 8 years on an investment of $6,000 with quarterly compounding?

```
                                          10,000
                                        ^  FV
                     i = ?              |
                                        |
 |-----|-----|--  ...  --|-----|-----|
 |                              n = 8 * 4
 v
 PV
-6,000
```

Keystrokes	Display		
[f][CLR FIN]			
8[END LINE]4[*][n]	**32.00**	**:F**	Calculates and stores n.
6000[CHS][PV]	**-6000.00**	**:F**	Stores PV (with minus sign for cash paid out).
10000[FV]	**10000.00**	**:F**	Stores FV.
[i]	**1.61**	**:F**	Periodic (quarterly) interest rate.
4[*]	**6.44**	**:F**	Annual interest rate.

Calculating the Present Value

1. Press [f][CLR FIN] to clear the financial registers.

2. Enter the number of payments or periods, using [n] or [12*].
3. Enter the periodic interest rate, using [i] or [12/].
4. Enter either or both of the following:
   * Payment amount, using [PMT].
   * Future value, using [FV].

   Note: Remember to observe the cash flow sign convention.
5. If a PMT was entered, press [f][BEG] or [g][END] to set the payment mode.
6. Press [PV] to calculate the present value.

Example 1: You're financing a new car purchase with a loan from an institution that requires 15% interest compounded monthly over the 4-year term of the loan. If you can make payments of $150 at the end of each month and your down payment will be $1,500, what is the maximum price you can pay for the car? (Assume the purchase date is one month prior to the date of the first payment.)

```
PV?                i = 15/12
 ^
 |
 |-----|-----|-- ... --|-----|-----|
       |     |         |     |     |
       v     v         v     v     v
      PMT                    n = 4 * 12
     -150
```

Keystrokes	**Display**		
[f][CLR FIN]			
4[g][12*]	**48.00**	**:F**	Calculates and stores n.
15[g][12/]	**1.25**	**:F**	Calculates and stores i.
150[CHS][PMT]	**-150.00**	**:F**	Stores PMT (with minus sign for cash paid out).
[g][END]	**-150.00**	**:F**	Sets payment mode to End.
[PV]	**5389.72**	**:F**	Maximum amount of loan.
1500[+]	**6889.72**	**:F**	Maximum purchase price.

Example 2: A development company would like to purchase a group of condominiums with an annual net cash flow of $17,500. The expected holding period is 5 years, and the estimated selling price at the time is $540,000. Calculate the maximum amount the company can pay for the condominiums in order to realize at least a 12% annual yield.

```
                    i = 12          540,000
         17,500                       FV
           PMT                        ^
            ^       ^       ^       ^      |
            |       |       |       |      |
      |-----|-----|-----|-----|-----|
      |                         n = 5
      |                         n = 5
      v
     PV?
```

Keystrokes	Display		
[f][CLR FIN]			
5[n]	**5.00**	**:F**	Stores n.
12[i]	**12.00**	**:F**	Stores i.
17500[PMT]	**17500.00**	**:F**	Stores PMT. Unlike in the previous problem, here PMT is positive since it represents cash *received.*
540000[FV]	**540000.00**	**:F**	Stores FV.
[g][END]	**540000.00**	**:F**	Sets payment mode to End.
[PV]	**-369494.09**	**:F**	The maximum purchase price to provide a 12% annual yield. PV is displayed with a minus sign since it represents cash paid out.

Calculating the Payment Amount

1. Press [f][CLR FIN] to clear the financial registers.
2. Enter the number of payments or periods, using [n] or [12*].
3. Enter the periodic interest rate, using [i] or [12/].
4. Enter either of the following:
   * Present value, using [PV].
   * Future value, using [FV].

   Note: Remember to observe the cash flow sign convention.
5. Press [f][BEG] or [g][END] to set the payment mode.
6. Press [PMT] to calculate the payment amount.

Example 1: Calculate the payment amount on a 29-year, $43,400 mortgage at 14 1/4% annual interest.

```
        43,400
          PV
          ^            i = 14.24/12
          |
          |
          |-----|-----|-- ... --|-----|-----|
                |     |         |     |     |
                v     v         v     v     v
              PMT?                   n = 29 * 12
```

Keystrokes	Display		
[f][CLR FIN]			
29[g][12*]	**348.00**	**:F**	Calculates and stores n.
14.25[g][12/]	**1.19**	**:F**	Calculates and stores i.
43400[PV]	**43400.00**	**:F**	Stores PV.
[g][END]	**43400.00**	**:F**	Sets payment mode to End.
[PMT]	**-523.99**	**:F**	Monthly payment (with
minus			
			sign for cash paid out).

Example 2: Looking forward to retirement, you wish to accumulate $60,000 after 15 years by making deposits in an account that pays 9 3/4% interest compounded semiannually. You open the account with a deposit of $3,200 and intend to make semiannual deposits, beginning six months later, from your profit-sharing bonus paychecks. Calculate how much these deposits should be.

```
                                                  60,000
                         i = 9.75/2             ^   FV
                                                |
                                                |
          |-----|-----|-- ... --|-----|-----|
          |     |     |         |     |     |
          |     v     v         v     v     v
          v   PMT?                    n = 15 * 2
          PV
        3,200
```

Keystrokes	Display		
[f][CLR FIN]			
15[END LINE]2[*][n]	**30.00**	**:F**	Calculates and stores n.
9.75[END LINE]2[/][i]	**4.88**	**:F**	Calculates and stores i.
3200[CHS][PV]	**-3200.00**	**:F**	Stores PV (with minus sign for cash paid out).
60000[FV]	**60000.00**	**:F**	Stores FV.
[g][END]	**60000.00**	**:F**	Sets payment mode to End.
[PMT]	**-717.44**	**:F**	Semiannual payment (with minus sign for cash paid out).

Calculating the Future Value

1. Press [f][CLR FIN] to clear financial registers.
2. Enter the number of payments or periods, using [n] or [12*].
3. Enter the periodic interest rate, using [i] or [12/].
4. Enter either or both of the following:

   * Present value, using [PV].
   * Payment amount, using [PMT].

   Note: Remember to observe the cash flow sign convention.

5. If a PMT was entered, press [f][BEG] or [g][END] to set the payment mode.
6. Press [FV] to calculate the future value.

Example 1: In Example 1 on page 48, we calculated that the payment amount on a 29-year, $43,400 mortgage at 14 1/4% annual interest is $523.99. If the seller requests a balloon payment at the end of 5 years, what would be the amount of the balloon?

```
43,400
  PV   ^
       |            i = 14.25/12
       |                                  n = 5 * 12
       |-----|-----|-- ... --|-----|-----|
             |     |         |     |     |
             v     v         v     v     v
            PMT                          v
          523.99                        FV?
```

Keystrokes	**Display**		
[f][CLR FIN]			
5[g][12*]	**60.00**	**:F**	Calculates and stores n.
14.25[g][12/]	**1.19**	**:F**	Calculates and stores i.
43400[PV]	**43400.00**	**:F**	Stores PV.
523.99[CHS][PMT]	**-523.99**	**:F**	Stores PMT (with minus sign for cash paid out).
[g][END]	**-523.99**	**:F**	Sets payment mode to End.
[FV]	**-42652.37**	**:F**	Amount of balloon payment.

Example 2: If you deposit $50 a month (at the beginning of each month) into a new account that pays 6 1/4% annual interest compounded monthly, how much will you have in the account after 2 years?

```
                                                   FV?
                                                    ^
                          i = 6.25/12               |
                                                    |
   |-----|-----|--  ...  --|-----|-----|
   |     |     |           |     |     |
   v     v     v           v     v     v
  PMT                                 n = 2 *12
   50
```

Keystrokes	Display		
[f][CLR FIN]			
2[g][12*]	**24.00**	**:F**	Calculates and stores n.
6.25[g][12/]	**0.52**	**:F**	Calculates and stores i.
50[CHS][PMT]	**-50.00**	**:F**	Stores PMT (with minus sign for cash paid out).
[f][BEG]	**-50.00**	**:F**	Sets payment mode to Begin.
[FV]	**1281.34**	**:F**	Balance after 2 years.

Example 3: Property values in an unattractive area are depreciating at the rate of 2% per year. Assuming this trend continues, calculate the value in 6 years of property presently appraised for $32,000.

```
                                                    FV?
                                                     ^
                          i = -2                     |
                                                     |
   |-----|-----|-----|-----|-----|-----|
   |
   |
   v
   PV                                  n = 6
-32,000
```

Keystrokes	Display		
[f][CLR FIN]			
6[n]	**6.00**	**:F**	Stores n.
2[CHS][i]	**-2.00**	**:F**	Stores i (with minus sign for a "negative interest rate").
32000[CHS][PV]	**-32000.00**	**:F**	Stores PV (with minus sign for cash paid out).
[FV]	**28346.96**	**:F**	Property value after 6 years.

Amortization

The C71 software enables you to calculate the amounts applied toward principal and toward interest from a single loan payment or from several payments, and also tells you the remaining balance of the loan after the payments are made.

> **NOTE:** All amounts calculated when [f][AMORT] is pressed are automatically rounded to the number of decimal places specified by the display format. This rounding affects the number inside the computer as well as how the number appears on the display. The amounts calculated may differ from those on the statements of lending institutions by a few cents, since different rounding techniques are sometimes used. For more information on rounding, refer to section 3, Display Control.

To obtain an amortization schedule:

1. Press [f][CLR FIN] to clear the financial registers.
2. Enter the periodic interest rate, using [i] or [12/].
3. Enter the amount of the loan (the principal), using [PV].
4. Key in the periodic payment, then press [CHS][PMT]. (The sign of PMT must be negative, in accordance with the cash flow sign convention.)
5. Press [f][BEG] or (for most direct reduction loans) [g][END] to set the payment mode.
6. Key in the number of payments to be amortized.
7. Press [f][AMORT] to display the amount from those payments applied toward interest.
8. Press [x<>y] to display the amount from those payments applied toward the principal.
9. To display the number of payments just amortized, press [v][v].
10. To display the remaining balance of the loan, press [RCL][PV].
11. To display the total number of payments amortized, press [RCL][n].

Example: For a house you're about to buy, you can obtain a 25-year mortgage for $50,000 at 13 1/4% annual interest. This requires payments of $573.35 (at the end of each month). Find the amounts that would be applied to interest and to the principal form the first year's payments.

Keystrokes	Display		
[f][CLR FIN]			Clears financial registers
13.25[g][12/]	**1.10**	:F	Enters i.
50000[PV]	**50000.00**	:F	Enters PV.
573.35[CHS][PMT]	**-573.35**	:F	Enters PMT (with minus sign for cash paid out).
[g][END]	**-573.35**	:F	Sets payment mode to End.
12[f][AMORT]	**-6608.89**	:F	Portion of first year's payments (12 months) applied to interest.
[x<>y]	**-271.31**	:F	Portion of first year's payments applied to principal.
[RCL][PV]	**49728.69**	:F	Balance remaining after 1 year.
[RCL][n]	**12.00**	:F	Total number of payments amortized.

The number of payments keyed in just before [f][AMORT] is pressed, is taken to be the payments following any that have already been amortized. Thus, if you now press 12 [f][AMORT], your computer will calculate the amounts applied to interest and to the principal from the second year's payments (that is, the second 12 months):

Keystrokes	Display		
12[f][AMORT]	**-6570.72**	:F	Portion of second year's payments applied to interest.
[x<>y]	**-309.48**	:F	Portion of second year's payments applied to principal.
[v][v]	**12.00**	:F	Number of payments amortized.
[RCL][PV]	**49419.21**	:F	Balance remaining after 2 years.
[RCL][n]	**24.00**	:F	Total number of payments amortized.

Pressing [RCL][PV] or [RCL][n] displays the number in the PV or n register. When you did so after each of the last two calculations, you may have noticed that PV and n had been changed from their original values. The computer does this so that you can easily check the remaining balance and the total number of payments amortized. But because of this, if you want to generate a new amortization schedule form the beginning, you must reset PV to its original value and reset n to 0.

For example, suppose you now wanted to generate an amortization schedule for each of the first two months:

Keystrokes	Display		
50000[PV]	**50000.00**	:F	Resets PV to original value.
0[n]	**0.00**	:F	Resets n to zero.
1[f][AMORT]	**-552.08**	:F	Portion of first payment applied to interest.
[x<>y]	**-21.27**	:F	Portion of first payment applied to principal.
1[f][AMORT]	**-551.85**	:F	Portion of second payment applied to interest.
[x<>y]	**-21.50**	:F	Portion of second payment applied to principal.
[RCL][n]	**2.00**	:F	Total number of payments amortized.

If you want to generate an amortization schedule but do not already know the monthly payment:

1. Calculate PMT as described under "Calculating the Payment Amount" earlier in this chapter.
2. Press 0 **[n]** to reset n to zero.
3. Proceed with amortization procedure listed under "Amortization" beginning with step 6.

Example: Suppose you obtained a 30-year mortgage instead of a 25-year mortgage for the same principal ($50,000) and at the same interest rate (13 1/4%) as in the preceding example. Calculate the monthly payment, then calculate the amounts applied to interest and to the principal form the first month's payments. Since the interest rate is not being changed, do not press **[f][CLR FIN]**; to calculate PMT, just enter the new value for n, reset PV, then press **[PMT]**.

Keystrokes	Display		
30[g][12*]	**360.00**	:F	Enters n.
50000[PV]	**50000.00**	:F	Resets PV.
[PMT]	**-562.89**	:F	Monthly payment.
0[n]	**0.00**	:F	Resets n to zero.
1[f][AMORT]	**-552.08**	:F	Portion of first payment applied to interest.
[x<>y]	**-10.81**	:F	Portion of first payment applied to principal.
[RCL][PV]	**49989.19**	:F	Remaining balance.

Section 7
Discounted Cash Flow: NPV and IRR

C71 provides functions for the two most widely-used methods of discounted cash flow analysis: [NPV] (net present value) and [IRR] (internal rate of return). These functions enable you to analyze financial problems involving cash flows (money paid out or received) occurring at regular intervals. As in compound interest calculations, the interval between cash flows can be any time period; however, the amount of these cash flows need not be equal. This is called uneven cash flow analysis.

To understand how to use [NPV] and [IRR], let's consider the cash flow diagram for an investment that requires an initial cash outlay (CF_0) and generates a cash flow (CF_1) at the end of the first year, and so on up to the final cash flow (CF_6) at the end of the sixth year. In the following diagram, the initial investment is denoted by CF_0, and is depicted as an arrow pointing down from the time line since it is cash paid out. Cash flows CF_1 and CF_4 also point down from the time line, because they represent projected cash flow losses.

```
                                   CF6
                 CF3         CF5     ^
           CF2     ^           ^     |
             ^     |           |     |
             |     |           |     |
 |-----|-----|-----|-----|-----|-----|
 |     |                 |
 |     v                 v
 v    CF1               CF4
CF0
```

NPV is calculated by adding the initial investment (represented as a negative cash flow) to the present value of the anticipated future cash flows. The interest rate, i, will be referred to in this discussion of NPV and IRR as the rate of return. The value of NPV indicates the result of the investment:

* If NPV is positive, the financial value of the investor's assets would be increased: the investment is financially attractive.
* If NPV is zero, the financial value of the investor's assets would not change: the investor is indifferent toward the investment.
* If NPV is negative, the financial value of the investor's assets would be decreased: the investment is not financially attractive.

A comparison of the NPV's of alternative investment possibilities indicates which of them is most desirable: the greater the NPV, the greater the increase in the financial value of the investor's assets.

IRR is the rate of return at which the discounted future cash flows equal the initial cash outlay: IRR is the discount rate at which NPV is zero. The value of IRR relative to the present value discount rate also indicates the result of the investment:

* If IRR is greater than the desired rate of return, the investment is financially attractive.

* If IRR is equal to the desired rate of return, the investor is indifferent toward the investment.

* If IRR is less than the desired rate of return, the investment is not financially attractive.

In uneven cash flow analysis, *cash flows always occur at the end of the period.* For example, if your problem has a cash flow occurring at the beginning of period 11, place the cash flow on the cash flow diagram at the *end* of period 10. The BEGinning/END of the period payment mode is not involved in uneven cash flow analysis.

Calculating Net Present Value (NPV)

Calculating NPV for Ungrouped Cash Flows.

If there are no equal consecutive cash flows, use the procedure described (and then summarized) below. With this procedure, NPV (and IRR) problems involving up to 20 cash flows (in addition to the initial investment CF_0) can be solved. If two or more consecutive cash flows are equal--for example, if the cash flows in periods three and four are both $8,500--you can solve problems involving more than 20 cash flows, or you can minimize the number of storage registers required for problems involving less than 20 cash flows, by using the procedure described under "Calculating NPV for Grouped Cash Flows".

C71 maintains a separate set of storage registers for storing the uneven cash flows. These registers are cleared upon pressing [f][CLR FIN]. They are also cleared when C71RAM is purged or a **Memory Lost** occurs.

The amount of the initial investment (CF_0) is entered into the computer using the [CF_0] key. Pressing [g][CF_0] stores CF_0 in storage register CF_0 and also stores the number 0 in the n register.

The amounts of the subsequent cash flows are stored--in the order they occur--in the remaining storage registers: CF_1 through CF_{20} in financial storage registers CF_1 through CF_{20}. Each of these cash flows (CF_1, CF_2, etc.) is designated CF_j, where j takes on values from 1 up to the number of the final cash flow. The amounts of these cash flows are all entered using the [CF_j] key. Each time [g][CF_j] is pressed, the amount in the display is stored in the next available register, and the number in the n register is increased by 1. This register therefore counts how many cash flow amounts (in addition to the initial investment CF_0) have been entered.

> **NOTE:** When entering cash flow amounts--including the initial investment CF_0--remember to observe the cash flow sign convention by pressing [CHS] after keying in a negative cash flow.

In summary, to enter the cash flow amounts:

1. Press [f][CLR FIN] to clear the financial storage registers.

2. Key in the amount of the initial investment, press [CHS] if that cash flow is negative, then press [g][CF_0]. If there is no initial investment, press 0 [g][CF_0].

3. Key in the amount of the next cash flow, press [CHS] if the cash flow is negative, then press [g][CF_j]. If the cash flow amount is zero in the next period, press 0 [g][CF_j].

4. Repeat step 3 for each cash flow until all have been entered.

With the amounts of the cash flows stored in the registers, you can calculate NPV as follows:

1. Enter the interest rate, using [i] or [12/].
2. Press [f][NPV].

The calculated value of NPV appears in the display and also is automatically stored in the PV register.

Example: An investor has an opportunity to buy a duplex for $80,000 and would like a return of at least 13%. He expects to keep the duplex 5 years and then sell it for $130,000; and he anticipates the cash flows shown in the diagram below. Calculate NPV to determine whether the investment would result in a return or a loss.

```
                                    130,000
                          5,500       CF5
                    4,500  CF3  4,500  ^
                     CF2    ^    CF4   |
                      ^     |     ^    |
                      |     |     |    |
      |-----|-----|-----|-----|-----|
      |  1  |  2    3     4     5
      |     v
      |    CF1
      v   -500
     CF0
   -80,000
```

Note that although a cash flow amount ($4,500) occurs twice, these cash flows are *not* consecutive. Therefore, these cash flows must be entered using the method described above.

Keystrokes	**Display**		
[f][CLR FIN]	**0.00**	:F	Clears financial and storage registers.
80000[CHS][g][CF0]	**-80000.00**	:F	Stores CF0 (with minus sign for a negative cash flow).
500[CHS][g][CFj]	**-500.00**	:F	Stores CF1 (with minus sign for negative cash flow).
4500[g][CFj]	**4500.00**	:F	Stores CF2.
5500[g][CFj]	**5500.00**	:F	Stores CF3.
4500[g][CFj]	**4500.00**	:F	Stores CF4.
130000[g][CFj]	**130000.00**	:F	Stores CF5.
[RCL][n]	**5.00**	:F	Checks number of cash flow amounts entered (in addition to CF0).
13[i]	**13.00**	:F	Stores i.
[f][NPV]	**212.18**	:F	NPV.

Since NPV is positive, the investment would increase the financial value of the investor's assets.

Calculating NPV for Grouped Cash Flows.

A maximum of 20 cash flow amounts (in addition to the initial investment CF_0) can be stored. However, problems involving more than 20 cash flows can be handled if among the cash flows there are *equal consecutive* cash flows. For such problems, you merely enter along with the amounts of the cash flows the number of times each amount occurs consecutively. This number is designated N_j, corresponding to cash flow *amount* CF_j, and is entered using the [Nj] key. Each N_j is stored in a special register inside the computer.

All values of N_j should be integers (no fractional periods) that are greater than zero. Noninteger values for any N_j will result in calculated numbers that are not meaningful in finance.

This method can of course, be used for problems involving fewer than 20 cash flows--and it will require fewer financial registers than the method described above under "Calculating NPV for Ungrouped Cash Flows". Equal consecutive cash flows can be entered using that method--provided the 20 financial registers available are enough to accommodate the total number of individual cash flows.

> **NOTE:** When entering cash flow amounts--including that of the initial investment CF_0--remember to observe the cash flow sign convention by pressing [CHS] after keying in the amount for a negative cash flow.

In summary, to enter the amounts of the cash flows and the number of times they occur consecutively:

1. Press [f][CLR FIN] to clear the financial storage registers.
2. Key in the amount of the initial investment, press [CHS] if that cash flow is negative, then press [g][CF_0]. If there is not initial investment, press 0 [g][CF_0].
3. If the initial investment consists of more than one cash flow of the amount entered in step 2, key in the number of those cash flows, then press [g][N_j]. If [g][N_j] is not pressed, C71 assumes that N_0 is 1.
4. Key in the amount of the next cash flow, press [CHS] if that cash flow is negative, then press [g][CF_j]. If the cash flow amount is zero in the next period, press 0 [g][CF_j].
5. If the amount entered in step 4 occurs more than once consecutively, key in the number of times that cash flow amount occurs consecutively, then press [g][N_j]. If [g][N_j] is not pressed, C71 assumes that N_j is 1 for the CF_j just entered.
6. Repeat steps 4 and 5 for each CF_j and N_j until all cash flows have been entered.

With the amounts of the cash flows and the number of times they occur consecutively stored in the computer, NPV can be calculated by entering the interest rate and pressing [f][NPV], just as described earlier.

Example: An investor has an opportunity to purchase a piece of property for $79,000; and he would like a 13 1/2% return. He expects to be able to sell it after 10 years for $100,000 and anticipates the yearly cash flow shown in the table below.

Year	Cash Flow	Year	Cash Flow
1	$14,000	6	$9,100
2	11,000	7	9,000
3	10,000	8	9,000
4	10,000	9	4,500
5	10,000	10	100,000

Since two cash flow amounts ($10,000 and $9,000) are repeated consecutively, we can minimize the number of financial registers required and the number of keystrokes required by using the method described.

Keystrokes	Display		
[f][CLR FIN]	**0.00**	**:F**	Clears financial and storage registers.
79000[CHS][g][CF$_0$]	**-79000.00**	**:F**	Initial investment (with minus sign for a negative cash flow).
14000[g][CFj]	**14000.00**	**:F**	First cash flow amount.
11000[g][CFj]	**11000.00**	**:F**	Next cash flow amount.
10000[g][CFj]	**10000.00**	**:F**	Next cash flow amount.
3[g][Nj]	**3.00**	**:F**	Number of times this cash flow amount occurs consecutively.
9100[g][CFj]	**9100.00**	**:F**	Next cash flow amount.
9000[g][CFj]	**9000.00**	**:F**	Next cash flow amount.
2[g][Nj]	**2.00**	**:F**	Number of times this cash flow amount occurs consecutively.
4500[g][CFj]	**4500.00**	**:F**	Next cash flow amount.
100000[g][CFj]	**100000.00**	**:F**	Final cash flow amount.
[RCL][n]	**7.00**	**:F**	Seven different cash flow amounts have been entered.
13.5[i]	**13.50**	**:F**	Stores i.
[f][NPV]	**907.77**	**:F**	NPV.

Since NPV is positive, the investment would increase the financial value of the investor's assets by $907.77.

Calculating Internal Rate of Return (IRR)

1. Enter the cash flows using either of the methods described under Calculating Net Present Value.

2. Press [f][IRR].

The calculated value of IRR appears in the display and also is automatically stored in the i register.

> **NOTE:** Remember that the [IRR] function may take a significant amount of time to produce an answer, during which the computer flashes **Working...** in the display.

Example: The NPV calculated in the preceding example was positive, indicating that the actual rate of return (that is, the IRR) was greater than 13 1/2% used in the calculation. Find the IRR.

Assuming the cash flows are still stored in the computer, we need only press [f][IRR]:

Keystrokes	Display		
[f][IRR]	**13.72**	**:F**	IRR is 13.72%

Note that the value calculated by [IRR] is the periodic rate of return. If the cash flow periods are other than years (for example, months or quarters), you can calculate the nominal annual rate of return by multiplying the periodic IRR by the number of periods per year.

As noted above, the computer may take several seconds or even a minute to produce an answer for IRR. This is because the mathematical calculations for finding IRR are extremely complex, involving a series of iterations--that is, a series of successive calculations. In each iteration, the computer uses an estimate of IRR as the interest rate in a computation of NPV. The iterations are repeated until the computed NPV reaches about zero. In practice, because the complex mathematical calculations inside the computer are done with numbers rounded to 12 digits, NPV may never reach exactly zero. Nevertheless, the interest rate that results in a very small NPV is very close to the actual IRR.

The complex mathematical characteristics of the IRR computation have an additional ramification: Depending on the magnitudes and signs of the cash flows, the computation of IRR may have a single answer, multiple answers, a negative answer or no answer.

The IRR is used in finance to make decisions about capital investments. To understand how it is used, we need to define the term *conventional investment.* If the cash flow series is structured from the viewpoint of the investor, a conventional investment is one in which all three of the following criteria are met:

* The first cash flows are negative and some cash flows come later that are positive.
* The sequence of cash flows changes sign (negative to positive) only once.
* The sum of all cash flows is positive.

If the investment proposal meets this definition, there is one and only one IRR.

For investments other than conventional investments, the decision rule may not apply. For the *nonconventional* investment (one that does not meet the three criteria) there are many possible situations. Refer to the practices and literature in your profession and industry to learn the applications and conventions used there.

If the program determines that the cash flow does not meet the above criteria an error message may be displayed indicating that the cash flows are zero or that no solution greater than zero exists. If it determines that multiple solutions exists, it will prompt for an estimate of i or the IRR.

The program uses a maximum of 15 iterations in searching for i%. If one is not found, the error message **Can't Solve** is displayed.

Advanced Users Note: The calculator starts with the number in *i* to start solving for IRR and i%. If you know about what the answer is going to be, you can reduce the calculating time by entering the approximate answer in *i*. If the message **Can't Solve** is displayed, it is possible that the beginning number in *i* was so far off, that the calculator could not converge on the answer. In that case try storing zero in *i* and try the calculation again. If the calculator finds zero in *i* when it starts, it will supply its own seed.

If you start with the internal seed, and get the message **Can't Solve**, you might try just pressing [IRR] or [i] again to see if it will converge from the result that is currently in *i*. The calculator updates *i* when it gets the **Can't Solve** error. Or if you think you know what the answer should be, you can try storing that in *i* before trying again. However, if the calculator could not solve the problem in 15 iterations starting from the internal seed, it is very likely that the problem is unsolvable.

Reviewing Cash Flow Entries

* Store the number of that cash flow amount (that is, the value of j for the CF_j desired) in the n register, then press [RCL][g][CFj].

* To review all the cash flow amounts, press [RCL][g][CFj] repeatedly. This displays the cash flow amounts in reverse order--that is, beginning with the final cash flow and proceeding to CF_0.

* To display the number of times a cash flow amount occurs consecutively--that is, to display the N_j for CF_j--store the number of that cash flow amount(that is, the value of j) in the n register, then press [RCL][g][Nj].

* To review all the cash flow amounts together with the number of times each cash flow amount occurs consecutively (that is, to review each CF_j and N_j pair), press [RCL][g][Nj][RCL][g][CFj] repeatedly. This displays N_j followed by CF_j beginning with the final cash flow amount and proceeding to N_0 and CF_0.

NOTE: Neither [IRR] nor [NPV] change the number in the n register. However, each time [RCL][g][CFj] is pressed, the number in the n register is decreased by 1. If this is done, or if you manually change the number in the n register in order to display a single N_j and/or CF_j, be sure to reset the number in the n register to the total number of cash flow amounts originally entered (not including the amount of the initial investment CF_0). If this is not done, NPV and IRR calculations will give incorrect results; also, a review of cash flow entries would begin with N_n and CF_n, where n is the number currently in the n register.

For example, to display the fifth cash flow amount and the number of times that amount occurs consecutively:

Keystrokes	Display		
5[n]	**5.00**	**:F**	Stores the value of j in the n register.
[RCL][g][Nj]	**2.00**	**:F**	N5.
[RCL][g]CFj]	**9000.00**	**:F**	CF5.
7[n]	**7.00**	**:F**	Resets the number in the n register to its original value.

To display all the cash flow amounts and the number of times they occur consecutively:

Keystrokes	Display		
[RCL][g][Nj]	**1.00**	**:F**	N7.
[RCL][g][CFj]	**100000.00**	**:F**	CF7.
[RCL][g][Nj]	**1.00**	**:F**	N6.
[RCL][g][CFj]	**4500.00**	**:F**	CF6.
[RCL][g][Nj]	**2.00**	**:F**	N5.
[RCL][g][CFj]	**9000.00**	**:F**	CF5.
.	.	.	.
.	.	.	.
.	.	.	.
[RCL][g][Nj]	**1.00**	**:F**	N1.
[RCL][g][CFj]	**14000.00**	**:F**	CF1.
[RCL][g][Nj]	**1.00**	**:F**	N0.
[RCL][g][CFj]	**-79000.00**	**:F**	CF0.
7[n]	**7.00**	**:F**	Resets the number in the n register to its original value.

Changing Cash Flow Entries

* To change a cash flow amount:

  1. Store the number of that cash flow amount (that is, the value of j) in the n register.
  2. Key the amount into the display.
  3. Press [STO][g][CFj].

* To change the number of times a cash flow amount occurs consecutively--that is, to change the Nj for a CFj:

  1. Store the number of that cash flow amount (that is, the value of j) in the n register.
  2. Key the number of times the cash flow amount occurs consecutively into the display.
  3. Press [STO][g][Nj].

Note: If you change the number in the n register in order to change a CF_j or an N_j, be sure to reset the number in the n register to the total number of cash flow amounts originally entered (not including the amount of the initial investment CF_0). If this is not done, NPV and IRR calculations will give incorrect results.

Example 1: With the cash flows now stored in the computer, change CF_2 from $11,000 to $9,000, then calculate the new NPV for a 13 1/2% return.

Keystrokes	Display		
2[n]	**2.00**	**:F**	
9000[STO][g][CFj]	**5.00**	**:F**	Stores the new CF_2 in R_2.
7[n]	**7.00**	**:F**	Reset n to its original value.
13.5[i]	**13.50**	**:F**	Stores i.
[f][NPV]	**-644.75**	**:F**	The new NPV.

Note: It was necessary in this example to reset n to its original value because we have calculated IRR since the first time we calculated NPV. The IRR calculation replaced the 13.5 we keyed into i before calculating NPV with the result for IRR--13.72.

Since this NPV is negative, the investment would decrease the financial value of the investor's assets.

Example 2: Change N_5 from 2 to 4, then calculate the new NPV.

Keystrokes	Display		
[5][n]	**5.00**	**:F**	Stores j in the n register.
4[STO][g][Nj]	**4.00**	**:F**	Stores the new N_5.
7[n]	**7.00**	**:F**	Resets the number in the n register to its original value.
[f][NPV]	-1857.21	:F	The new NPV.

Section 8
Statistics Functions

Random Number Generator

C71's random number generator uses the HP-71's internal random number generator. The seed used is the seed that the HP-71 currently has stored. On initial power-on or on **Memory Lost** the seed is set to 999500333083533. To set the seed to some other value, use the BASIC keyword in the basic system **RANDOMIZE.**

To get the next random number in the sequence press **[f][RAN#]**. The new random number will appear in the displayed X-register. (**[f][RAN#]** affects the stack in the same way as recalling a number from a storage register.) The random number will be a number in the range 0 <= r < 1. Also the random number seed is updated for calculating the next random number.

Accumulating Statistics

C71 can perform one- or two- variable statistical calculations. The data is entered into the computer using the [Σ+] key, which automatically calculates and stores statistics of the data into the internal HP-71 statistics storage array.

Before beginning to accumulate statistics for a new set of data, you should clear the statistics array by pressing **[f][CLR STAT]**. If the buffer does not exist, that is, no statistics have yet been entered, the calculator will beep and display **Invalid Stat Array.** This is not an error to be concerned with, it merely means that you have tried to clear a nonexistent statistics array. The next time you press [Σ+], a statistics array will be created.

In one-variable statistical calculations, to enter each data point--referred to as "coordinate value 1"--key coordinate value 1 into the display, then press [Σ+].

In two-variable statistical calculations, to enter each data pair--referred to as "coordinate value 1 and coordinate value 2":

1. Key coordinate value 2 into the display.
2. Press **[END LINE]**.
3. Key coordinate value 1 into the display.
4. Press [Σ+]. The current number of accumulated data points, *n*, will be displayed. The *x*-value is saved in the LAST X register and *y* remains in the Y-register. [Σ+] disables stack lift, so the stack will not lift when the next number is keyed in.

Refer to "Mathematical Discussion of HP-71 Statistical Arrays" on page 334 in the HP-71 Reference Manual for additional information on how the HP-71 maintains the statistical array.

Correcting Accumulated Statistics

If you discover you have entered data incorrectly, the accumulated statistics can easily be corrected:

* If the incorrect data point or data pair has just been entered and [Σ+] has been pressed, press [f][LX][f][Σ-].

* If the incorrect data point or data pair is not the most recent one entered, key in the incorrect data point or data pair again as if it were new, press [f][Σ-] instead of [Σ+].

These operations cancel the effect of the incorrect data point or data pair. You can then enter the data correctly, using [Σ+], just as if it were new.

MEAN

Pressing [MEAN] calculates the mean (*arithmetic averages*) of the coordinate value in the displayed X-register. The mean of the specified coordinate value appears in the display after [MEAN] is pressed. To display the mean of coordinate value 1, press 1[f][MEAN]. To display the mean of coordinate value 2, press 2[f][MEAN].

Example: A survey of seven salespersons in your company reveals that they work the following hours a week and sell the following dollar volumes each month. How many hours does the average salesperson work each week? How much does the average salesperson sell each month?

Salesperson	Hours/Week	Sales/Month
1	32	$17,000
2	40	$25,000
3	45	$26,000
4	40	$20,000
5	38	$21,000
6	50	$28,000
7	35	$15,000

To find the average workweek and sales of this example:

Keystrokes	Display		
[f][CLR STAT]	**0.00**	**:F**	Clears stat registers.
32[END LINE]	**32.00**	**:F**	
17000[Σ+]	**1.00**	**:F**	First entry.
40[END LINE]	**40.00**	**:F**	
25000[Σ+]	**2.00**	**:F**	Second entry.
45[END LINE]	**45.00**	**:F**	
26000[Σ+]	**3.00**	**:F**	Third entry.
40[END LINE]	**40.00**	**:F**	
20000[Σ+]	**4.00**	**:F**	Fourth entry.
38[END LINE]	**38.00**	**:F**	
21000[Σ+]	**5.00**	**:F**	Fifth entry.
50[END LINE]	**50.00**	**:F**	
28000[Σ+]	**6.00**	**:F**	Sixth entry.
35[END LINE]	**35.00**	**:F**	
15000[Σ+]	**7.00**	**:F**	Total number of entries in the sample.
1[f][MEAN]	**21714.29**	**:F**	Mean dollar sales per month (coordinate 1).
2[f][MEAN]	**40.00**	**:F**	Mean workweek in hours (coordinate 2).

Standard Deviation

Pressing [SD] calculates the standard deviation of the coordinate value specified in the displayed X-register. (The standard deviation of a set of data is a measure of the dispersion around the mean.) The standard deviation of the specified coordinate value appears in the display after [SD] is pressed. To display the standard deviation of the coordinate 1 value, press 1[g][SD]. To display the standard deviation of the coordinate 2 value, press 2[g][SD].

Example: To calculate the standard deviation of the x-values and of the y-values from the preceding example:

Keystrokes	Display		
1[g][SD]	**4820.59**	**:F**	Standard deviation of sales.
2[g][SD]	**6.03**	**:F**	Standard deviation of hours worked.

The formulas used for calculating the standard deviation give *best estimate* of the population standard deviation based on a sample of the population. Thus, current statistical convention calls it the *sample* standard deviation. So we have assumed that the seven salespersons are a sample of the population of *all salespersons*, and our formulas derive best estimates of the population from the sample.

What if the seven salespersons constituted the whole population of salespersons. Then we wouldn't need to *estimate* the *population* standard deviation. We can find the *true population* standard deviation ( ) when the data set equals the total population, using the following keystrokes.

Keystrokes	Display		
2[f][mean]	**40.00**	**:F**	Mean workweek hours.
1[f][mean]	**21714.29**	**:F**	Mean (dollars).
[Σ+]	**8.00**	**:F**	Number of entries +1.
1[g][SD]	**4463.00**	**:F**	of coordinate value 1.
2[g][SD]	**5.58**	**:F**	of coordinate value 2..

To continue summing data pairs, press **2[f][mean]1[f][mean][f][Σ-]** before entering more data.

Linear Estimation

With two-variable statistical data accumulated in the statistics registers, you can estimate a new coordinate 2 value given a new coordinate 1 value, and estimate a new coordinate 1 value given a new coordinate 2 value.

Because the HP-71 can allow 15 coordinate values (refer to the Advanced User information in this section), a dependent and independent variable must first be specified to the linear regression function. The number of the coordinate value that is the dependent variable--the value that will be predicted--is placed in the Y-register. The number of the coordinate value that is the independent variable--the value you choose--is placed in the X-register. Pressing [g][LR], sets the linear regression model and returns the intercept of the line in the Y-register and the slope of the line in the X-register.

As normally only 2 coordinate values will be used (that is all that can be entered using [Σ+]), if you want to predict a new coordinate 2 value, 2 must be placed in the Y-register and 1 in the X-register. If you want to predict a new coordinate 1 value, place a 1 in the Y-register and a 2 in the X-register.

To calculate a new coordinate 2 value:

1. Press **2[END LINE]1[g][LR]** to set the linear regression model. The intercept is returned in the Y-register and the slope is in the X-register.
2. Key in a new independent variable value.
3. Press **[f][P.VAL]**.

To calculate a new coordinate 1 value:

1. Press **1[END LINE]2[g][LR]** to set the linear regression model. The intercept is returned in the Y-register and the slope is in the X-register.
2. Key in a new independent variable value.
3. Press **[f][P.VAL]**.

Example: Using the accumulated statistics from the preceding problem, estimate the amount of sales delivered by a new salesperson working 48 hours per week.

Keystrokes	Display		
2[END LINE]1[g][LR]	720.18	:F	The slope of the regression line.
[X<>Y]	-7093.05	:F	The intercept.
48[f][P.VAL.]	27475.75	:F	Estimated sales for a 48-hour workweek.

Correlation Coefficient

The reliability of a linear estimate depends upon how closely the data pairs would, if plotted on a graph, lie in a straight line. The usual measure of this reliability is the correlation coefficient, r. This quantity is calculated by placing the number of one of the coordinate values in the X-register and the number of the other coordinate values in the Y-register. It is not important which number is placed in the X-register or the Y-register. So, for the 2 coordinate values that can be entered with [Σ+], press **1[END LINE]2[f][CORR]** or **2[END LINE]1[f][CORR]**.

A correlation coefficient close to 1 or -1 indicates that the data pairs lie very close to a straight line. On the other hand, a correlation coefficient close to 0 indicates that the data pairs do not lie closely to a straight line; and a linear estimate using this data would not be very reliable.

Example: Check the reliability of the linear estimate in the preceding example by displaying the correlation coefficient.

Keystrokes	**Display**		
1[END LINE]2[f][CORR]	0.90	:F	The correlation coefficient is close to 1, so the sales calculated in the preceding example is a good estimate.
2[END LINE]1[f]CORR]	0.90	:F	The correlation coefficient is the same for both ways of specifying the coordinate values.

TOTAL

[TOTAL] returns the sum of the data elements in the specified coordinate value. Pressing **1[g][TOTAL]** returns to the X-register the sum of the data in coordinate value 1. Pressing **2[g][TOTAL]** returns to the X-register the sum of coordinate value 2. Pressing **0[g][TOTAL]** returns to the X-register the number of data points in the array.

Example: Find the sum of the data elements in the preceding example:

Keystrokes	**Display**		
1[g][TOTAL]	**152000.00**	**:F**	Sum of the Sales/Month.
2[g][TOTAL]	**280.00**	**:F**	Sum of the Hours/Week.
0[g][TOTAL]	**7.00**	**:F**	7 data points were entered into the arrays.

Advanced User's Information

The internal HP-71 statistical formulas can use statistical arrays of up to 15 coordinate values. C71 is structured to allow you to use the full array capability of the HP-71's functions with one important limitation:

> The [Σ+] and [Σ-] can be used to enter only a 2 coordinate value array. The array uses R7(2) as the 2 dimensional variable used by the HP-71's internal statistical functions.

Only the **[f][CLR STAT]**, the [Σ+], and the [f][Σ-] keys expect the two dimensional array R7. All other functions will directly use the currently declared statistical array.

To enter a larger number of coordinate values into an array, exit the calculator by pressing **[g][EXIT]**. Follow the instructions in "Section 4: Statistical Functions" in the HP-71 Owner's Manual to declare and create a statistical array using **STAT**, **CLSTAT**, **ADD**, and **DROP**. When all of the data values have been entered, return to the calculator by pressing **C71**.

From C71 [MEAN], [SD], [CORR], [TOTAL], [LR], and [P.VAL] can all be used by placing the appropriate number of the coordinate values in the X- and Y-register. For example if a statistical array of 10 coordinate values has been created, then to get the correlation coefficient between the fifth and ninth coordinate values, you would press **5[END LINE]9[f][CORR]**.

Section 9

Complex Number Functions

C71 enables you to calculate with complex numbers, that is, numbers of the form

a + ib

where a is the real part of the complex number,

b is the imaginary part of the complex number, and

$i = \sqrt{-1}$

Complex Numbers and the Stack

C71 uses the X-, Y-, Z-, and T-registers to form a pair of complex number registers. The X- and Y-register form the real and complex parts of the first complex number respectively, while the Z- and T-registers form the real and complex parts of the second number.

T	t	iz_y	
Z	z	z_x	z register
Y	y	iw_y	
X	x	w_x	w register
	Regular Stack	Complex Stack	

The stack operates in a similar fashion to the standard memory stack, except that only two numbers may be represented. A complex number is entered into the z- and w-registers with the result being returned in the w-register.

Entering Complex Numbers

To enter a number with real and imaginary parts:

1. Key the complex part of the number into the display.
2. Press [END LINE].
3. Key the imaginary part of the number into the display.
4. After the operation, the real part is in the display.
4. Press [X<>Y] to see the complex result.

Example: Add 2+3i and 4+5i.

Keystrokes	**Display**		
[f][FIX]4			
3[END LINE]	**3.0000**	**:F**	Keys imaginary part of first number into display.
2[END LINE]	**2.0000**	**:F**	Keys real part of first number into display.
5[END LINE]	**5.0000**	**:F**	Keys imaginary part of second number into display.
4	**4**	**:F**	Keys real part of second number into display.
[f][C+]	**6.0000**	**:F**	Displays real part of sum.
[X<>Y]	**8.0000**	**:F**	Displays imaginary part of sum.

The stack operations looked like this (assuming the stack was cleared with [CLR STK]):

(T)		0.0000	0.0000	0.0000	3.0000	3.0000
(Z)	z ->	0.0000	0.0000	3.0000	2.0000	3.0000
(Y)		0.0000	3.0000	2.0000	5.0000	8.0000
(X)	w ->	3	2	5	4	6.0000
		3[END LINE]	2[END LINE]	5[END LINE]	4	[f][C+]

Operations with Complex Numbers

One-Number Functions

The following functions operate on both the real and imaginary parts of a single complex number in the w-register and return the result to the w-register.

[SINZ], [COSZ], [TANZ], [|Z|], [1/Z], [e^Z], [LNZ]

[|Z|] returns the magnitude of the w-register to the X-register--there is no imaginary part.

For the tignonometric functions, the calculator considers the complex number in the w-register to be expressed in radians--regardless of the current trigonometric mode.

These numbers are entered by:

1. Key the complex part into the display.
2. Press [END LINE].
3. Key in the real part into the display.
4. Press the execute of the function.

5. The real part is displayed in the X-register.
6. Press [X<>Y] to view the complex part.

Example: Find [SINZ] of the complex number (2 + 3i).

Keystrokes	**Display**		
3[END LINE]	**3.0000**	**:F**	Complex part of number.
2	**2**	**:F**	Real part of number.
[SINZ]	**9.1545**	**:F**	Real part of result
[X<>Y]	**-4.1689**	**:F**	Complex part of result

Complex Arithmetic Functions

The following functions operate on two complex numbers, one in the w-register and one in the z-register. The result is returned to the w-register.

[C+], [C-], [C*], [C\], [Z^W], [Z^1/W]

These numbers are entered by:

1. Key the complex part of z into the display.
2. Press **[END LINE]**.
3. Key in the real part of z into the display.
4. Press **[END LINE]**.
5. Key the complex part of w into the display.
6. Press **[END LINE]**.
7. Key in the real part of w into the display.
8. Press the execute of the function.
9. The real part is displayed in the X-register.
10. Press [X<>Y] to view the complex part.

Example: Evaluate

$$\frac{Z_1}{Z_2 + Z_3}$$

where $Z_1 = 23 + 13i$, $Z_2 = -2 + i$, and $Z_3 = 4 - 3i$. Since the functions can remember only two numbers at a time, perform the calculation as:

$Z_1 \times [1/(Z_2 + Z_3)]$.

Keystrokes	Display		
1[END LINE]	**1.0000**	**:F**	Complex part of Z_2.
2[CHS][END LINE]	**-2.0000**	**:F**	Real part of Z_2.
3[CHS][END LINE]	**-3.0000**	**:F**	Complex part of Z_3.
4	**4**	**:F**	Real part of Z_3.
[f][C+]	**2.0000**	**:F**	Real part of the result of $(Z_2 + Z_3)$.
[X<>Y]	**-2.0000**	**:F**	Complex part of result.
[X<>Y]	**2.0000**	**:F**	Remember to restore the result to proper order for the next calculation.
[1/Z]	**0.2500**	**:F**	Real part of $1/(Z_2 + Z_3)$.
[X<>Y]	**0.2500**	**:F**	Complex part of result.
[X<>Y]	**0.2500**	**:F**	Restore proper order.
13[END LINE]	**13.0000**	**:F**	Complex part of Z_1.
23	**23**	**:F**	Real part of Z_1.
[f][C*]	**2.5000**	**:F**	Real part of final result.
[X<>Y]	**9.0000**	**:F**	Complex part of result.

One Complex Number Mixed with Real Arguments

One Real Argument. The following functions operate on a single complex number in the z-register and one real number in the X-register.

<u>[Z^n]</u>, <u>[a^z]</u>, <u>[LOG$_a$Z]</u>

where: n is an integer, and
a is real.

These numbers are keyed in by:

1. Key the complex part into the display.
2. Press <u>[END LINE]</u>.
3. Key in the real part into the display.
4. Press <u>[END LINE]</u>.
5. Key the real number n or a into the display.
6. Press the execute of the function.
7. The real part is displayed in the X-register.
8. Press <u>[X<>Y]</u> to view the complex part.

Example: The characteristic impedance of a ladder network is given by an equation of the form

$$Z_0 = \sqrt{\frac{A}{B}} = (A/B)^{.5}$$

where A and B are complex numbers. Find Z_0 for the values A = 1.2 + 4.7i and B = 2.7 + 3.2i.

Keystrokes	**Display**		
4.7[END LINE]	**4.7000**	**:F**	Complex part of A.
1.2[END LINE]	**1.2000**	**:F**	Real part of A.
3.2[END LINE]	**3.2000**	**:F**	Complex part of B.
2.7	**2.7**	**:F**	Real part of B.
[f][C/]	**1.0428**	**:F**	Real part of A/B.
.5	**.5**	**:F**	Power for square root.
[Z^n]	**1.0491**	**:F**	Real part of Z_0.
[X<>Y]	**0.2406**	**:F**	Complex part of Z_0.

Two Real Arguments. The following function operates on one complex number in the z-register and two real arguments in the X- and Y-registers.

[Z^1/n]

where: n is an integer.

[Z^1/n] will have n roots--you must specify which root to calculate:

1. Key the complex part into the display.
2. Press [END LINE].
3. Key in the real part into the display.
4. Press [END LINE].
5. Key the real number n into the display.
6. Press [END LINE].
7. Key in the number of the root. If the root number is not less than the root n, then an **Invalid Arg** error message is displayed.
8. Press the execute of the function.
9. The real part is displayed in the X-register.
10. Press [X<>Y] to view the imaginary part.

Complex numbers have as many roots as the number of the root. Root are numbered k = 0 through k = n - 1. A square root has two roots--k = 0 and k = 1. A cube root it is three--k = 0, k = 1, and k = 2. This pattern is continued for higher order roots. In the previous example by using **[Z^n]** with n = .5, we forced only the 0th root which did not allow for specifying which root we wanted. The proper method would be to use **[Z^1/n]** and specify the desired root.

Example: Redo the previous example specifying the second root, k = 1. This time the equation is

$Z_0 = (A/B)^{1/2}$, at the k=1 root.

Keystrokes	Display		
4.7[END LINE]	**4.7000**	**:F**	Complex part of A.
1.2[END LINE]	**1.2000**	**:F**	Real part of A.
3.2[END LINE]	**3.2000**	**:F**	Complex part of B.
2.7	**2.7**	**:F**	Real part of B.
[f][C/]	**1.0428**	**:F**	Real part of A/B.
2[END LINE]	**2.0000**	**:F**	Specifies square root.
1	**1**	**:F**	The k = 1 root.
[Z^1/n]	**-1.0491**	**:F**	Real part of the second root of Z_0.
[X<>Y]	**-0.2406**	**:F**	Complex part of the second root of Z_0.

Section 10

Computer Scientist Basics

This section will discuss the different aspects of integer number use and display: number bases, word size, complements, number ranges, and the resulting displays found in the Computer Scientist mode of the C71 Calculator. Floating-point format is in Engineering/Business mode and is described in the previous sections.

Integer Mode

The number base modes ([HEX], [DEC], [OCT], and [BIN]) operate strictly in Computer Scientist or Integer mode (that is, using integers only). Fractional decimal numbers can be used in Engineering/Business mode. Pressing [f][CS] while in Engineering/Business mode will change the mode from Engineering/Business or Floating Point mode to Computer Scientist or Integer mode.

The memory stack and LAST X register all work exactly the same as described in section 2, The Automatic Memory Stack, LX, and Data Storage.

Number Base Modes

There are four number base modes used by C71 in Integer mode for the purposes of display and digit entry. Hexadecimal (base 16), Decimal (base 10), Octal (base 8), and Binary (base 2). An **H**, **D**, **O**, or **B** to the right of the colon (**:**) on the display indicates the present number base mode.

Pressing [HEX], [DEC], [OCT], or [BIN] converts the display to that number base in a right-justified, integer format. Digit keys pressed are interpreted accordingly: the calculator will not respond if you attempt to enter an inappropriate digit (such as a "3" in Binary mode), Hexadecimal mode uses the keys [A] to [F], appearing in the display as **A**, **B**, **C**, **D**, **E**, and **F**.

> **NOTE:** Regardless of the current number base mode, the *internal representation of numbers is always binary*. Switching between number modes changes the display only, not the calculator's internal representation of the value.

Complement Modes and Unsigned Mode

The C71 provides three conventions for representing numbers: *1's Complement* mode, *2's Complement* mode, and *Unsigned* mode. The 2's Complement mode is the default mode when the C71 software is 'first' run. Once a mode is set, it remains in effect until you change it or until C71RAM is purged. (All examples in the Computer Scientist sections use 2's Complement unless otherwise indicated.)

In the binary representation of a signed number, the leftmost or most significant bit with respect to word size serves as the sign bit: 0 for plus and 1 for minus. In Decimal mode, a negative number is displayed with a minus sign.

1's Complement Mode

Pressing [f][1's] will set 1's Complement mode. When you press [CHS] (change sign) in 1's Complement mode, the 1's complement of the number in the X-register is formed by complementing all bits.

One's Complement accommodates an equal number of positive and negative numbers, but has two representations for zero: 0 and -0.

2's Complement Mode

Pressing [f][2's] will set 2's Complement mode. The [CHS] function will take the 2's complement of the number in the display (that is, it complements all the bits in the X-register and adds 1).

In 2's Complement there is just one representation for zero, but there is always one more negative number than positive number represented.

Unsigned Mode

Pressing [f][UNS] will set Unsigned mode, which uses no sign bit. The most significant bit adds magnitude, not sign, so the largest value represented by an 8-bit word is 255_{10} instead of 127_{10}.

Changing signs in Unsigned mode has no meaning. If you press [CHS] in Unsigned mode, the result will be the 2's complement of the number in the X-register. Flag 4 is set as a reminder that the true result is a negative number, which is outside the range of Unsigned mode.

The following table summarizes how the complement modes affect the decimal interpretation of all possible 4-bit patterns (word size 4).

Decimal Interpretation of 4-bit Binary

Binary	1's Complement Mode	2's Complement Mode	Unsigned Mode
0111	7	7	7
0110	6	6	6
0101	5	5	5
0100	4	4	4
0011	3	3	3
0010	2	2	2
0001	1	1	1
0000	0	0	0
1111	-0	-1	15
1110	-1	-2	14
1101	-2	-3	13
1100	-3	-4	12
1011	-4	-5	11
1010	-5	-6	10
1001	-6	-7	9
1000	-7	-8	8

Word Size and Window Display

C71 will work with *words* (data units) up to 64 bits long. The default *word size* when you enter Computer Scientist mode the very first time or C71RAM is purged is 16 bits. The display window shows 16 digits at a time; leading zeros are not displayed (unless flag 2 is set--described later). The left and right arrows at the top left- and top right-hand side of the display respectively, indicate the presence of more, undisplayed digits to the left or right of the current displayed portion of a number.

Word Size

To specify a word size, first place the desired word size (1_{10} to 64_{10}) in the X-register, then press [f][WSIZE]. The absolute value of the number is used; a zero is interpreted as 64. After [WSIZE] is executed the stack drops.

A current word size smaller than 8 will limit the size of the number you can enter to stipulate a new word size; but you can always enter [0][f][WSIZE] to set a word size of 64. (You can then set any word size.) You will get an error message "**Too Large...**" if you attempt to specify a word size larger than 64.

The calculator will not allow you to enter more digits than allowed by the word size.

Keystrokes	**Display**	
[DEC] 16 [f][WSIZE]		Base 10; word size 16.
[f][2's]		Sets 2's Complement mode.
32767 [END LINE]	**32767 :D**	Largest positive 2's complement number with a word size of 16.
8 [f][WSIZE]	**-1 :D**	Number changes from 01111111 11111111 (base 2, 16 bits) to 11111111 (base 2, eight bits).
16 [f][WSIZE]	**255 :D**	Number changes from 11111111 (base 2) to 00000000 11111111.

> NOTE: A change in word size might not preserve numerically equivalent values stored *in the memory stack*. Going to a smaller word size will truncate a word, leaving the least significant bits. Going to a larger word size will not preserve the sign bit of a negative number. If the original word size is restored, the original stack contents are not restored.

Windows

The display can be considered a *window* showing up to sixteen digits of the number in the X-register. The X-register, like all *registers*, can hold up to 64 binary digits, depending on the word size. What you normally see is window 0, the 16 least significant digits of the number in the X-register. As you key in more than 16 digits, the most significant digits move off the left end of the display into window 1.

Pressing [f][WINDOW] {0 to 3} will display different 16-digit portions of the word in the X-register. The display returns to window 0, the sixteen least significant digits of the word, with each new entry into the X-register. The highest window number is 3 since the maximum word size is 64. (With smaller word sizes or smaller numbers, the higher windows will be blank.) **Invalid Window** error message is displayed if you specify a window number greater than 3.

Example: The 16-digit hexadecimal value FF00 FF00 FF00 FF00 has 64-digit binary representation (eight 1's alternating with eight 0's). In Binary mode, you can view the entire number by executing [f][WINDOW]0 through [f][WINDOW]3.

```
  X-register:

     1111111100000000 1111111100000000 1111111100000000
11111110000000

         Window 3          Window 2          Window 1
Window 0

(default)

     most significant <----------------------------> least
significant
```

Scrolling

Scrollng with the [SCRL.L] and [SCRL.R] keys allows you to move different parts of a number into the display, one digit at a time. This does not change the number itself, only what part of the number you see.

The arrows tells you where to look for the rest of the number in the X-register. For instance, if the *left arrow* is displayed, then there are more digits to the *left* of the current display. Pressing [SCRL.R] will scroll the number to the *right*, bringing these "hidden" digits into view. Both arrows may be displayed, indicating that the window is not at one end of the number.

Example: The following scrolling and [WINDOW] functions can be used to view the entire X-register contents. The word size used is 20 bits.

Keystrokes	**Display**	
[BIN]		Sets Binary mode.
1111111111111111	1111111111111111 :B	Display filled (16 digits).
[END LINE]	1111111111111111 :B	
1[+]	<0000000000000000 :B	Arrow on the left side, so the number continues to the left.
[SCRL.R]	1000000000000000 :B>	Scrolls number one digit to right (right arrow shows number now continues to the right).
[f][WINDOW]1	1 :B>	The contents of window 1: the most significant digit.

[f][WINDOW]0	<0000000000000000 :B	Window 0: the least significant digits.

Scrolling is "reset"--that is, the display is reset to window 0--when a bit manipulation or mathematical function is executed.

The Display and Internal Representation

The following keystrokes illustrate how various functions (number base, word size, complement mode) alter the calculator's display in relation to the internal binary representation

Keystrokes	**Display**	Internal Binary Representation
[HEX]		
8 [f][WSIZE]		
[<]	**0 :H**	00000000
62	**62 :H**	01100010
[OCT]	**142 :O**	01100010
[BIN]	**1100010 :B**	01100010
[DEC]	**98 :D**	01100010
62	**62 :D**	00111110
[OCT]	**76 :O**	00111110
62	**62 :O**	00110010
[HEX]	**32 :H**	00110010
[f][2's]	**32 :H**	00110010
[CHS]	**CE :H**	11001110 Negative number
in		
		2's Complement, word size 8.
[OCT]	**316 :O**	11001110
[BIN]	**11001110 :B**	11001110
[DEC]	**-50 :D**	11001110
[f][1's]	**-49 :D**	11001110 Internal representation does not change.
[f][UNSGN]	**206 :D**	11001110
[f][1's]	**-49 :D**	11001110 In 1's
Complement		
		this is interpreted as a negative number.
[CHS]	**49 :D**	00110001
2	**2 :D**	00000010
5	**25 :D**	00011001
4	**-1 :D**	11111110 (Corresponds to -1 base 10 in 1's Compl.
[f][UNSGN]	**254 :D**	11111110

Flags

C71 uses three flags--displayed as flags 2, 3, and 4--which are used to indicate system status.

* Flag 2 controls the display of leading zeros. When it is set, zeros to the left of the highest nonzero digit are displayed up to the number of digits allowed by word size. When it is clear (the default condition), the display of leading zeros is suppressed. (Note that leading zeros are always suppressed in Decimal mode).

* Flag 3 is set when a **carry** has occurred.

* Flag 4 is set when the returned value is **out-of-range** (greater than the largest representable number or not representable in the current mode).

Refer to the discussions in section 11 "Arithmetic and Bit Manipulation Functions" for how the carry and out-of-range conditions are generated.

The flags can be set, cleared, and tested as follows:

* [g][SF] *n* will *set flag* number *n* (2, 3, or 4);
* [g][CF] *n* will *clear flag* number *n*; and
* [g][F?] *n* will check if flag *n* is set.

  Note: Flags 0 and 1 are used in Engineering/Business mode and cannot be set, cleared, or tested while in Computer Scientist mode. Also flags 0 and 1 will not be displayed (but will not be changed) while in Computer Scientist mode. Any flag specified other than 2, 3, or 4 will cause a **Not a flag...** error message to be displayed.

A flag's status is retained until changed by:

* Purging C71RAM or a **Memory Lost** condition.

* Executing a function which affects that flag (flag 3 and 4 only).

* Clearing the flag with [CF] or setting it with [SF].

The function [F?] just displays a **Yes...** or **No...** depending on the state of the flag.

Machine Status

Pressing [f][STATUS] will temporarily show the current word size and the current complement. The display remains as longs as you hold down the [STATUS] key. To alter the machine status, refer to Complement Modes and Word Size.

The status display will look as follows:

    WS: 16 COMPL: U

In this example the word size is sixteen and the complement is unsigned.

The current flag settings are continuously displayed in the far right-hand column--flags 2, 3, and 4. When the annunciator is on, the flag is set. When the annunciator is off, the flag is clear.

The Effects of Switching Modes

When [f][EB] is pressed, the X-, Y-, Z-, T-, and Last X registers are cleared, the status unique to the Computer Scientist mode is stored, the previously stored Engineering/Business status is restored, and an **:F** is displayed in the far left of the display. Upon returning to Computer Scientist, the last previously stored status will be restored. However, the stack will be cleared.

The status items that are stored and then restored are:

* Wordsize.
* Complement mode.
* Flags 2, 3, and 4.

The storage registers are not cleared. However, numbers stored in Engineering/Business or Floating Point mode are of a different format than the numbers stored in Computer Scientist mode. So a floating point number recalled while in Computer Scientist mode will not be converted or be the same. The calculator will try and interpret the bit pattern in the register by the format rules of Computer Scientist. If it is unable to make any type of conversion a **Data Type** error message will be displayed. Refer to *Number Formats* in section 12, Interface to BASIC, for additional details.

Storage Registers

The storage registers R_0 through R_9 are shared by the Engineering/ Business calculator and the Computer Scientist calculator. As mentioned above the numbers stored in one mode will not be usable in the other mode. The registers are always 64-bits wide and are not adjusted by changing the word size. The numbers in the registers are not converted or sign extended when word size or complement modes are changed. When a number larger than the current word size is recalled, the number is truncated to the current wordsize. It is not sign compressed. When a number smaller than the current wordsize is recalled, no change in the number is made. It also is not sign extended. A number stored will be the same number and sign if it is recalled with the same wordsize even if there has been a different word size in affect between storage and recall.

Section 11
Arithmetic and Bit Manipulation Functions

Integer arithmetic operations and bit manipulation functions can only be performed in Integer mode. Since these functions are subject to carry and out-of-range conditions, an explanation of these conditions precedes the discussion of the functions themselves.

Floating-point decimal arithmetic and other capabilities are described in the Engineering/Business sections earlier in this manual.

Carry and Out-of-Range Conditions

The execution of certain arithmetic and bit manipulation operations can result in a *carry* and/or an *out-of-range* condition. These conditions set flags (that may be tested) and display annunciators (for visual indication). The definitions for "carry" and "out-of-range" depend on the particular function executed.

Section 10, under Flags, explains how to manually set and clear these and other flags.

Flag 3: Carry

The shifting, rotating, and arithmetic operations listed below will set or clear flag 3 whenever they are performed in Integer mode. Flag 3, the carry flag, is set if the carry bit is 1, and cleared if the carry bit is 0.

```
[SL]      [RL]      [RLn]       [+](carry)
[SR]      [RLC]     [RLCn]      [-](borrow)
[ASR]     [RR]      [RRn]       [/](remainder <> 0)
          [RRC]     [RRCn]
```

Through the remainder of this section, all examples will show a status line to the right of the keystroke example. This shows what wordsize, complement mode, and flag settings in the calculator must be set to at the start of the example. The flag line is blank when no flags are set (as the flags are blank in the display). If a flag is set, its number will be shown next to flags as follows: Flags: 2 (indicates that flag 2 is set).

Example: The following simple additions set and then clear the carry flag (3).

Keystrokes	**Display**	**([STATUS]: WS: 16 Compl: 2)**
		Flags:
[HEX] FFFF [END LINE]	**FFFF :H**	Hex mode
1 [+]	**0 :H**	Carry occurred and flag 3 set.
1 [+]	**1 :H**	Carry cleared because no carry occurred.

Flag 4: Out-of-Range

Flag 4 is set if the correct result of an operation cannot be represented in the current word size and complement mode. For the [+] and [-] operations, this corresponds to the "overflow" condition on most computers.

The functions below either set or clear flag 4 whenever they are performed in Integer mode:

[+] [-] [*] [/] [ABS] [CHS]

When a result is out-of-range, the lower bits (as many as fit in the given word size) of the full answer will be returned. If the operation was [*] or [/] in 1's or 2's Complement mode, the most significant bit (sign bit) returned will match the sign bit of the full answer.

Keystrokes	**Display**	**([STATUS]: WS: 16 Compl: 2) Flags:**
[DEC] 32767 [END LINE]	**32767 :D**	
2 [*]	**32766 :D**	Flag 4 set for overflow. Leading binary digit is zero; number is positive.
[g][CF] 4	**32766 :D**	Clears flag 4.

Arithmetic Functions

Addition, Subtraction, Multiplication, and Division

The arithmetic operations [+], [-], [*], and [/] can be performed using integers in any of the four number bases. The operands, which can be entered in different bases, must be in the Y and X-registers. After the operation is performed, the stack drops and the result is placed in the X-register.

In Integer mode, [/] performs an integer division. The fractional part of the quotient is truncated.

All the arithmetic operators except [*] will set or clear flag 3 and flag 4 whenever executed. [*] affects flag 4 only.

Example: Find $(5A0_{16}) / (177764_8)$.

Keystrokes	**Display**	**([STATUS]: WS: 16 Compl: 2) Flags:**
[HEX] 5A0 [END LINE]	**5A0 :H**	Enters first number.
[OCT] 177764	**177764 :O**	Changes to Octal; keys in second number.
[/]	**177610 :O**	Result in base 8. Since a carry was not generated, the result is exact.
[HEX]	**FF88 :H**	Converts to base 16.

Addition and Subtraction in 1's Complement Mode. In 2's Complement and Unsigned modes, the result of an addition or subtraction is simply the difference of the two bit patterns in the X- and Y-registers. In 1's Complement mode, however, the result of an addition is affected by the occurrence of a carry, and the result of a subtraction is affected by the occurrence of a borrow. If a carry out of the most significant bit occurs, 1 is added to the result. If a borrow into the most significant bit occurs, 1 is subtracted from the result. Both cases set flag 3.

`([STATUS]: WS: 4 Compl: 1 Flags: 2)`

Carry

$$\begin{array}{r} -1 \\ +(-1) \\ \hline -2_{10} \end{array} \qquad \begin{array}{r} 111 \\ 1110 \\ +1110 \\ \hline 1100 \\ +\quad 1 \\ \hline 1101_2 \end{array}$$

No Carry

$$\begin{array}{r} -3 \\ +3 \\ \hline -0_{10} \end{array} \qquad \begin{array}{r} 1100 \\ +0011 \\ \hline 1111_2 \end{array}$$

Borrow

$$\begin{array}{r} 3 \\ -4 \\ \hline -1_{10} \end{array} \qquad \begin{array}{r} 11 \\ \cancel{0}011 \\ -0100 \\ \hline 1111 \\ -\quad 1 \\ \hline 1110_2 \end{array}$$

No Borrow

$$\begin{array}{r} 6 \\ -5 \\ \hline 1_{10} \end{array} \qquad \begin{array}{r} 0110 \\ -0101 \\ \hline 0001_2 \end{array}$$

The Carry Flag During Addition. The carry flag (flag 3) will be set whenever a binary addition results in a carry "out of" the most significant bit. If an addition does not result in such a carry, the carry flag is cleared. This is the same for all complement modes.

`([STATUS]: WS: 4 Compl: 2 Flags: 2)`

Carry Set

$$\begin{array}{r} -6 \\ +(-4) \\ \hline 6_{10} \end{array} \qquad \begin{array}{r} 1 \\ 1010 \\ +1100 \\ \hline 0110_2 \end{array}$$

(Incorrect, so out-of-range flag set also)

Carry Cleared

$$\begin{array}{r} 6 \\ +1 \\ \hline 7_{10} \end{array} \qquad \begin{array}{r} 0110 \\ +0001 \\ \hline 0111_2 \end{array}$$

The Carry Flag During Subtraction. The carry flag (flag 3) will be set whenever a binary subtraction results in a borrow into the most significant bit. Otherwise, the carry flag is cleared. This is the same for all complement modes. (Note that subtraction in C71 is *not* computed as the addition of a negative number; this affects how carry generation occurs.)

([STATUS]: WS: 4 Compl: 2 Flags: 2)

Carry Set		Carry Cleared	
-6	$\overset{0\,1}{\cancel{1}}010$	6	$01\overset{0\,1}{\cancel{1}}0$
$-(-4)$	-1100	-1	-0001
	-----		-----
-2_{10}	1110_2	5_{10}	0101_2

The Out-of-Range Flag. Arithmetic results that cannot be shown in the current word size and complement mode set the out-of-range flag. For [/], this occurs only in 2's Complement mode when the largest possible negative number is divided by -1.

Example: With a word size of 4 bits, calculate (7 + 6) in base 2 and observe the effect on flags 3 and 4.

Keystrokes **2)**	**Display**	**([STATUS]: WS: 4 Compl:** **Flags:**
[BIN]		Binary mode.
111 [END LINE]	**111 :B**	7.
110	**110 :B**	6.
[+]	**1101 :B**	-3. Flag 4 (out-of-range) set; flag 3 (carry) cleared.

Remainder After Division and [RMD]

In division, only the integer portion of the result is returned to the X-register. If the remainder is not zero, flag 3 (carry) is set. If the remainder is zero, flags 3 is cleared.

To obtain the remainder instead of the quotient, press [f][RMD] instead of [/]. This calculates |y|MOD|x|. The sign of the result matches the sign of the dividend (that is, the sign of y).

Keystrokes **Compl: 2)**	**Display**	**([STATUS]: WS: 16** **Flags:**
[HEX] 66 [END LINE]	**66 :H**	Hexadecimal mode.
7 [/]	**E :H**	Flag 3 is set. 66/7 leaves a remainder.
2 [/]	**7 :H**	Flag 3 cleared. E/2 leaves no remainder.
4 [f][RMD]	**3 :H**	Remainder of 7/4.

Negative Numbers and Complementing

Changing signs. The [CHS] function (*change sign*) will change the sign, forming the complement (1's or 2's) of the number in the X- register. If the X-register holds the largest possible negative number in 2's Complement mode, the only effect of pressing [CHS] will be to set flag 4 (out-of-range).

In Unsigned mode, [CHS] forms a 2's complement and sets flag 4 as a reminder that a negative number is outside the range of Unsigned mode.

To key in a negative number, press [CHS] *after* its digits have been keyed in. [CHS] terminates digit entry in Integer mode.

Absolute Value. Pressing [f][ABS] converts the number in the X- register to its absolute value, forming the 1's or 2's complement of a negative number. There is no change if the calculator is in Unsigned mode or if the number is positive.

If the X-register holds the largest possible negative number in 2's Complement mode, the only effect of [ABS] will be to set flag 4 (out-of-range).

Logical Operations

The logical (Boolean) operations NOT, OR, AND, and EXCLUSIVE OR return the results of a bit-by-bit analysis of a binary number. The functions [OR], [AND], and [XOR] operate on the bits in corresponding positions of the X and Y-registers; the stack then drops and the result is placed in the X-register. The operator [NOT] acts only upon the word in the X-register; the stack does not drop.

NOT

The [NOT] function inverts the values of all bits in the binary number in the X-register. It is equivalent to forming the 1's complement, that is, using [CHS] in 1's Complement mode. Only the X-register is affected.

Keystrokes	**Display**	**([STATUS]: WS: 16 Compl: 2)** **Flags:**
[BIN] 11111111	**11111111 :B**	Binary mode.
[f][NOT]	**1111111100000000 :B**	1's complement of 00000000 11111111

AND

The [AND] function (the *logical product*) compares each corresponding bit in two words. Each resulting bit is 1 only if both corresponding operand bits are 1; otherwise, it is 0.

The use of [AND] is illustrated under Masking, page 95.

OR

The [OR] function (the *logical sum*) compares each corresponding bit in two words. Each resulting bit is 0 only if both operand bits are 0's.

Example: Perform a logical OR to determine which bits are zero in both 10101 and 10011 (base 2).

Keystrokes	**Display**	**([STATUS]: WS: 16 Compl: 2) Flags:**
10101 [END LINE]	**10101 :B**	
10011	**10011 :B**	
[f][OR]	**10111 :B**	Bit 3 (represented by the zero) and all bits to the left of bit 4 are zero in both of the given words.

EXCLUSIVE OR

The [XOR] function (the *logical difference*) compares the corresponding bits in two words and yields a 1 only if two corresponding bits are different.

Example: Use the [XOR] function to determine if two binary quantities (01010101_2 and 01011101_2) are the same. A 1 in the result signifies that the two quantities are different at those bit position(s).

Keystrokes	**Display**	**([STATUS]: WS: 16 Compl: 2) Flags:**
1010101 [END LINE]	**1010101 :B**	
1011101	**1011101 :B**	
[f][XOR]	**1000 :B**	The two numbers differ in the fourth bit from the right.

Shifting and Rotating Bits

Shifting and rotating operations cause the bits of a word to be moved left or right. The fate of the bit moved off the end of the word, and the value of the bit entering the vacated position, depend upon the type of shift or rotate performed.

Flag 3 (carry) is set or cleared by any shift or rotate function, except [LJ] (*left-justify*), as shown in the diagrams below.

Shifting Bits

The C71 software can perform two types of shifts on the contents of the X-register: a *logical shift* or an *arithmetic shift*. The latter preserves the sign bit. The contents of the X-register can also be *left-justified*.

Logical Shifts. Pressing [SL] (*shift left*) or [SR] (*shift right*) moves all the bits of the word in the X-register one bit to the left or right. Bits shifted out of the word are shifted into the carry bit, and the previous state of the carry bit is lost. The new bits generated at the opposite end of the word are always zeros.

```
[SL]      [ ] <-- [<----------*] <-- 0
         Carry      X-register

[SR]       0 -->  [*---------->] --> [ ]
                    X-register      Carry
```

Left-Justify. To left-justify a bit pattern within its word size, press [LJ]. The stack will lift, placing the left-justified word in the Y-register and the count (number of bit-shifts necessary to left-justify the word) in the X-register. The carry flag is not affected by [LJ].

Example: Left-justify the binary value 1111 in a word size of eight.

Keystrokes	**Display**	**([STATUS]: WS: 8 Compl: 2) Flags:**
1111	**1111 :B**	
[LJ]	**100 :B**	The count: four bit-shifts to left justify the word.
[v]	**11110000 :B**	Left-justified word.

Arithmetic Shift Right. Pressing [ASR] (*arithmetic shift right*) will move the contents of the word in the X-register one bit to the right, as does [SR]. However, instead of placing a zero into the new place at the left of the word, *the sign bit is regenerated.* (In Unsigned mode, which has no sign bit, [ASR] operates like [SR].) The carry bit is set if a 1 is shifted out of the X-register and cleared if 0 is shifted out.

```
[ASR]       [ *------------>] -----> [ ]
              ^
              |Sign Bit Unchanged      C
```

Example: Shifting a positive binary number to the right n places is equivalent to dividing it by 2^n. Since it regenerates the sign bit, an arithmetic shift also can be used to divide an even negative integer by 2. Divide 01111111 (word size 8) by 2^3, then divide 10000000 by 2^3.

Keystrokes 2)	**Display**	**([STATUS]: WS: 8 Compl: Flags:**
[g][SF]2		Allows display of leading zeros.
1111111	**01111111 :B**	
[DEC]	**127 :D**	
[BIN]	**01111111 :B**	
[SR][SR][SR]	**00001111 :B**	Each shift performs an integer divide by 2 and sets flag 3 because a 1 is shifted into the carry bit.
[DEC]	**15 :D**	
[BIN]	**00001111 :B**	
10000000	**10000000 :B**	
[DEC]	**-128 :D**	
[BIN]	**10000000 :B**	
[ASR][ASR][ASR]	**11110000 :B**	Sign bit is regenerated and carry flag is cleared with each shift.
[DEC]	**-16 :D**	
[BIN]	**11110000 :B**	

Rotating Bits

There are three general types of rotate functions in C71, encompassing eight different functions.

* Rotate left and right ([RL],[RR]).
* Rotate left and right "through the carry bit" ([RLC], [RRC]).
* Rotate *n* places ([RLn], [RRn], [RLCn], [RRCn]).

Rotation. Pressing [RL] (*rotate left*) or [RR] (*rotate right*) causes the contents of the X-register to rotate (or "circularly shift") one bit to the left or right. Bits shifted out of the word re-enter it at the other end. The carry flag is set if a 1 bit is rotated around the end, and is cleared if a zero is rotated around the end.

```
[RL]        [ ] <------- [<-----------------*]
             C           |__________________^|

[RR]        [*----------------->] -------> [ ]
             ^|__________________|           C
```

Rotation Through the Carry Bit. The [RLC] and [RRC] (*rotate left through carry* and *rotate right through carry*) functions respectively load the leftmost or rightmost bit of a word into the carry bit, and move the carry bit into the other end of the word.

```
[RLC]        [ ] <-------  [<-----------------*]
              |                               ^
              |_______________________________|
              C

[RRC]        [*----------------->]  -------> [ ]
              ^                               |
              |_______________________________|
                                              C
```

Rotating More Than One Bit at a Time. Given a bit pattern in the Y-register and *n* in the X-register, pressing [RLn], [RRn], [RLCn], or [RRCn] will rotate the pattern |*n*| bits. The stack drops, placing the result in the X-register.

The status of the carry flag (flag 3) is the same as if [RL], [RR], [RLC], or [RRC] were performed |*n*| times. For instance, executing [RRn] with n=3 will set the carry flag only if the third bit from the right (bit 2) is 1.

Example: Develop a keystroke sequence that will serve to rotate left as one word a 16-bit word *divided into two separate 8-bit words* held in two separate registers. For instance, with a word size of eight bits, rotate the word 00011100 11100111.

Keystrokes	**Display**	**([STATUS]: WS: 8 Compl: 2) Flags: 2**
11100	**00011100 :B**	High order portion of 16-bit word.
[SL]	**00111000 :B**	Moves most significant bit into carry bit (0 is moved into carry).
[f][LX]	**00011100 :B**	Recovers high-order portion.
11100111	**11100111 :B**	Low-order portion of 16-bit word.
[RLC]	**11001110 :B**	Carry bit (most significant bit of high-order portion) moves into least significant bit position of low-order portion.
[x<>y]	**00011100 :B**	Switches X and Y-registers.
[RLC]	**00111001 :B**	Carry bit moved into first part of word and zero moved into carry bit.
[x<>y]	**11001110 :B**	New word is 00111001 11001110.
[g][CF] 2	**11001110 :B**	Suppresses leading zeros.

Setting, Clearing, and Testing Bits

Individual bits in a word can be set to 1 or cleared to 0 using the [SB] (*set bit*) and [CB] (*clear bit*) functions. In a manner analogous to flag-testing, you can also test for the presence of a set bit with [B?].

To set, clear, or test a specific bit in a word:

* The word containing the specific bit must be in the Y- register.
* The magnitude of the number in the X-register specifies the number of the bit to be set, cleared, or tested.

When the key ([SB] or [CB]) is pressed, the stack drops and the word affected returns to the X-register.

Bits are numbered from zero to one less than the word size, with the least significant bit as bit number 0.

Keystrokes	**Display**	**([STATUS]: WS: 16 Compl: 2) Flags:**
11111111 [END LINE]	**11111111 :B**	Enters quantity and copies it into the Y-register.
11	**11 :B**	Bit number 3.
[f][CB]	**11110111 :B**	Stack drops; resulting word is in X-register.

[B?] tests a bit for a set or clear condition. If the bit is set, **Yes...** is displayed. If the bit is clear, **No...** is displayed. Press any key to return the display to its previous setting.

Masking

The [MASKL] (*mask left*) and [MASKR] (*mask right*) functions create left- or right-justified strings of 1 bits. The magnitude of the number in the X-register is used to determine how many 1's will comprise the mask. Upon execution, the mask pattern is placed in the X-register (the stack does not move).

You can create a mask as large as the word size. To place a mask in the middle of the field of a number, use a shift function in conjunction with [MASKL] or [MASKR].

Example: The ASCII representation of a two-digit number occupies 16 bits--eight bits per digit. Given an ASCII "65" (0011 0110 0011 0101). extract the high-order digit (6), thereby converting half of this ASCII code to binary coded decimal.

```
      0011 0110 0011 0101  ASCII "65"("3","6","3","5").
[AND] 0000 1111 0000 0000  Mask.
      0000 0110 0000 0000  The extracted, high-order digit ("6").
```

You can save keystrokes in this example by shifting the digits into position before masking.

Keystrokes	Display	([STATUS]: WS: 16 Compl: 2) Flags:
[HEX] 3635 [END LINE]	**3635 :H**	
8 [RRn]	**3536 :H**	Rotates word eight bits to the right to right-justify the desired hex digit (6).
4 [MASKR]	**F :H**	Right-justifies a mask of four 1 bits (1111) in the 16-bit word.
[f][AND]	**6 :H**	Extracts the rightmost four bits (6).

Bit Summation

Pressing **[f][#B]** (*number of bits*) sums the bits in the X-register and returns that value to the X-register. The bit pattern is saved in the LX register. No stack lift occurs. (In word sizes 1 and 2, the result must be interpreted in Unsigned mode.)

Section 12
Interface to BASIC

C71 provides two functions to allow you to interact with the BASIC operating system. These functions allow you to use numbers stored in the calculator within BASIC programs or user defined functions. Also this allows you to use feature of other plug-in modules that extend the operating features of the HP-71 such as the HP 82480A Math Pac.

NOTE

This section presumes that the user has a good foundation in the following areas of the HP-71 BASIC operating system:

* Programming
* File formats
* User-defined functions
* File structure of the HP-71
* Using data files

This section is not a tutorial, but is designed for the advanced user. For information regarding these topics, refer to the appropriate sections of the HP-71 Owner's Manual and the HP-71 Reference Manual.

The two basic interface functions are **[BFP]** (*BASIC Floating Point*) and **[BX]** (*BASIC Execute).* **[BFP]** allows computation of BASIC numerical expressions and user-defined functions where only the X-register is involved. **[BX]** is used for BASIC statements, running programs, and where more than one register value is needed for the calculation. **[BFP]** is available only in floating point (Engineering/Business) mode. **[BX]** is available to both Engineering/Business and Computer Scientist modes.

The following paragraphs describe the interactions in more detail. The last topic entitled "Additional Information" details the things to be careful of while using these functions. Additional information can also be found in Appendix D, HP-71 System Usage.

BASIC Floating Point Interaction

[BFP] interfaces to the basic interpreter for simple BASIC numerical expressions like (12.4 x 13.2)^3.5 / (sin(56) + 3) and user-defined functions like FNP(45.7). The result is placed in the X-register. The stack is not lifted, but the old value of the X-register is placed in the LAST-X.

Getting Values from C71

The only calculator register available is the X-register through the **SFORTHF** BASIC key word. Examples of **SFORTHF** are:

X=SFORTHF	Copies the contents of the X-register into the BASIC variable X.

R=SIN(SFORTHF)	Computes the sine of the contents of the X-register and places the result into the BASIC variable R.

Any operation that does not result in a single numeric result using BASIC functions will cause a BASIC system error. All BASIC statements will cause the error message **ERR:Data Type** to be reported and no other actions will be taken. Only floating point numbers are usable and therefore this function is available only in the Engineering/Business calculator.

Retrieving Values from BASIC

A numeric expression or function will always have some result that will be displayed if executed from the keyboard except for assignments: A=45. The result from what would normally be displayed if executed from the keyboard, will be captured by C71 and returned automatically to the X-register. Assignments or user-defined functions that have no return will by default return a zero. Thus something will always be returned except if an error occurs or nothing is typed prior to pressing [END LINE].

Keyboard and Display Interaction

When you press [f][BFP], a colon prompt (:) will be placed in the display with the cursor turned on and next to the prompt. All of the calculator-defined and user-defined keys will be released. The keyboard will be restored to the same state it was in upon entering the calculator. This means that if you had the USER keys active, they will be active again. The normal line editing keys will also be active so you can insert, delete, or otherwise edit the line until the [END LINE] key is pressed. [END LINE] terminates the key input and automatically causes the BASIC interpreter to evaluate what was typed in.

The keystrokes will be evaluated and the result will be placed into the X-register. The stack will not be lifted. The original X will be placed in LAST-X.

Example. The following is an example of how to use [BFP] with user-defined functions to find the circumference or area of a circle whose radius is in the X-register.

Exit the calculator by pressing [g][EXIT].

From BASIC press **EDIT TEST** to edit the test program we will use for this example. Now enter the following program:

```
100 ! TESTUDEF program -- User-definitions for BFP example
110 !
120 DEF FNC ! Circumference user-defined function
130 D=2*SFORTHF
140 FNC=D*PI ! Return the value
150 END DEF
160 !
170 DEF FNA ! Area user-defined function
180 FNA=PI*SFORTHF^2 ! Return value
190 END DEF
```

Now re-enter the calculator by pressing **C71**.

Keystrokes	Display		
20	**20**	**:F**	Radius of first circle.
[f][BFP]	**:**		Colon prompt.
FNC	**:FNC**		User-defined function name for finding the circumference.
[END LINE]	**125.6637**	**:F**	The circumference.
[f][LX]	**20.0000**	**:F**	Get radius back.
[f][BFP]	**:**		Colon prompt.
FNA	**:FNA**		User-defined function name for finding the area.
[END LINE]	**1256.6371**	**:F**	The area.

BASIC Execute Interaction

[BX] interfaces to the basic interpreter for BASIC statements like **EDIT TMPFILE**, or **RUN VECTPROG.**

Exchanging Data between C71 and BASIC

[BX] allows BASIC to access to all of C71's stack and storage registers by creating a data file in an HP-AF format and writing all of the convertible registers into the data file as strings. The file is created in main memory and named C71FILE. If a file by that name already exists, an error message--**ERR:File Exists**--will be displayed, the function will end, and the calculator will be active again. Data lines that are not convertible are written as empty strings. Each line in the data file contains one register converted into a string. The format of the data file is:

Record#	Contents
0	HPAFS
1	15
2	18
3	[Last X]
4	[X]
5	[Y]
6	[Z]
7	[T]
8	[R0]
9	[R1]
10	[R2]
11	[R3]
12	[R4]
13	[R5]
14	[R6]
15	[R7]
16	[R8]
17	[R9]
18	{'FP' | 'CS',Word size,Base}

Record 0 is a string indicating this is an HP-AF file with one string record per data line or record. Record 1 is a numerical entry indicating that there are 15 records of data, however all of the records do not actually need to contain data, some may be empty strings. Record 2 is a numerical entry indicating that record number 18 contains custom descriptor information for this file. This record--18--contains the string **FP** if the numbers were put in each record as floating point number strings. Otherwise, this record contains three fields: The first is the string **CS** meaning that the numbers are in CS format. The second field is a numerical entry containing the word-size that was in effect when the file was created and the data written. The third field is a numerical entry containing the Base that was used to convert the numbers into strings. The actual strings of data start in record 3 and continue through record 17.

[BX] allows basic programs to be run or basic statements to be executed. If a numeric expression is typed in, the expression will be evaluated, displayed at the delay rate, but will not be returned to the calculator.

Upon completion of the commands typed in, the calculator will become active again and will try to recover any data in the C71FILE data file. The calculator will look for a file named C71FILE and will try to assign a channel number to it. If one is not found, the BASIC system will create it upon channel number assignment. C71 will then try to read the first record which is nonexistent, which will cause an error--**ERR:End of File**--to be displayed. No data will be changed.

If the file with the name C71FILE is found, but is not a data, text, or sdata type file, then an error--**ERR:Invalid File Type**--will be displayed and no data will be changed.

When the data file is read by the calculator, record 0 is checked for the string HPAFS, record 1 is checked for the number 15, and record 2 is checked for the number 18. Record 18 is ignored other than to verify that a either the string FP or CS is in the first field depending on the state of the calculator. The calculator tries to reconvert the data into internal calculator representations according to the mode (ie. floating point or computer science) it was in when it created the file. The calculator will not change modes between FP and CS due to the contents of record 18. Nor will it change Word-size or Base due to the contents of record 18.

If anything different is in these records, then the calculator will display **ERR:Invalid File Type** error message, will not update or change any registers, will not purge C71FILE, and will return to the active calculator. C71FILE is not purge so you can examine its contents and work with your BASIC program to read and write the correct records and formats. In short this provides you with a debugging capability. However, remember that if you do not purge C71FILE prior to executing another [BX], the calculator will display **ERR:File Exists** and will terminate the function.

The calculator will then convert and store all possible records into their respective register. Any line that it cannot convert will be ignored and that register will not be changed. Empty strings are used as an indicator not to change the respective register.

The C71FILE is a data type file created as a random access file. It is 1292 bytes long and the system requires some additional bytes in order to create it. If not enough free memory is available to create the file, an **Insufficient Memory** error is displayed and control is returned to the calculator.

The file is created using the following BASIC statement:

CREATE DATA C71FILE,19,68

This format allows 64 bit numbers in base 2 to be stored. All registers occupy one line each.

Number Formats

C71 uses different internal number formats for floating point numbers in Engineering/Business mode and integers in Computer Scientist mode. A number stored in one format may or may not be similar enough to the other number format to be convertible into a string. If it is converted into a string, that string will not have the same value as the register originally contained.

Floating Point Format. Floating point numbers are stored in the calculator in the same format as the HP-71 BASIC system internally stores them. Numbers are stored as BCD (Binary Coded Decimal) as follows:

- 123456789012 499

The first field is the sign of the mantissa, followed by the 12 digits of the mantissa. The last three digits are the exponent stored as complement 10's mode. This is the internal CPU register format. The sequence is reversed if viewed from memory. There are no actual spaces in the number stored, this was done just for ease of reading.

This value is converted into a string by the equivalent of **STR$**. So the number gets written to the C71FILE as a string formatted according to the current display format. Excess digits of the mantissa are rounded according to the rounding mode set in the BASIC system.

Computer Scientist Format. Numbers are stored in the Computer Scientist as right justified numbers as they are seen in the display. They are stored strictly as binary data for the 64 bits in the register. (The format below is the CPU register format; the sequence would be reversed if viewed from memory.) So 5 would be stored as:

```
Bit Number ->   64 63    ...    3 2 1 0

  Contents ->    0  0           0 1 0 1
```

The values are converted into a string by the same function that performs the display formatting. So the string is truncated to word size, written from high order digits to low order digits, with the base determining the actual digits used. While the string is left justified in the register, only actual displayed digits will be written. So the above string would be only 3 digits long in base 2 and appear in a record as:

101

The trailing 61 places in the string will be nulls. If flag 2 is set, then the appropriate number of leading 0's will also be written. Remember these are ASCII characters that are written into the string and not binary data. The above string is the ASCII character 1 (decimal 49), the ASCII character 0 (decimal 48), and the ASCII character 1 (decimal 49).

Converting Mismatched Number Formats. As you can see from the above discussion, it is easy to create numbers in Computer Scientist mode that cannot be converted into a valid floating point number. The least significant bits in CS mode occur at the exponent of a floating point number. However, it is almost always true that the Computer Scientist can make some kind of conversion on a floating point number, even if its just to convert the exponent digits.

Keyboard and Display Interaction

When you press [f][BX], the calculator displays "**Working...**" while it is creating and writing to the C71FILE. It will take several seconds to create the file and write all the data. When the file has been completed, a colon prompt (**:**) will be placed in the display with the cursor turned on and next to the prompt. All of the calculator defined and user-defined keys will be released. The keyboard will be restored to you in the same state it was in upon entering the calculator. This means that if you had the USER keys active they will be active again. The normal line editing keys will also be active so you can insert, delete, or otherwise edit the line until the [END LINE] key is pressed. [END LINE] terminates key input and automatically causes the BASIC interpreter to evaluate what was typed in.

The line typed in will be executed. Upon termination of all the activity specified by the line, control will be returned to the calculator. The calculator will read the file and update the registers.

[BX] in Engineering/Business Mode

When the calculator is in Engineering/Business mode, the numbers are converted into a string equivalent number like STR$ does. To convert the number back into a number that calculations can be performed on, use the VAL function. Then when you are ready to return the value to the C71FILE, convert it back to a string using the STR$ function.

If a value or a string is written to the file that cannot be converted into a valid number by VAL from STR$, then the number will not be converted and the corresponding register will not be changed.

Floating Point Example

This example takes a radius in the X-register and a length in the Y-register and calculates various areas and volumes and stores the results in registers R_0 through R_5. Specifically:

R_0 -- The circumference of the circle.
R_1 -- The area of the circle.
R_2 -- The surface area of the sphere.
R_3 -- The volume of the sphere.
R_4 -- The surface area of the cylinder.
R_5 -- The volume of the cylinder.

Enter the following program as **TESTFP** into your HP-71.

```
100 ! This is the TESTFP program to illustrate BX in Foating Point mode
110 !
120 DIM A$[64]
130 REAL A,R,L,R0
140 ON ERROR GOTO 'FILERR'
150 !
160 ! Open the file
170 ASSIGN #1 TO C71FILE
180 !
190 ! Check for valid HP-AF format
200 READ #1,0;A$
210 IF A$#"HPAFS" THEN GOTO 'FILERR'
```

```
220 READ #1,1;A
230 IF A#15 THEN GOTO 'FILERR'
240 READ #1,2;A
250 IF A#18 THEN GOTO 'FILERR'
260 READ #1,18;A$
270 IF A$#"FP" THEN GOTO 'FILERR'
280 !
290 ! Now calculate values
300 READ #1,4;A$ ! Get radius from X
310 R=VAL(A$) ! Turn it into a number
320 READ #1,5;A$ ! Get length
330 L=VAL(A$) ! Turn it into a number
340 R0=2*PI*R ! circumfrence of circle
350 A$=STR$(R0) ! Turn it into a string
360 PRINT #1,8;A$ ! Put it into register R0
370 A$=STR$(PI*R^2) ! String value of area of circle
380 PRINT #1,9;A$ ! Put it into register R1
390 PRINT #1,10;STR$(4*PI*R^2) ! Put surface area of sphere in R2
400 PRINT #1,11;STR$(4/3*PI*R^3) ! Volume of sphere in R3
410 PRINT #1,12;STR$(2*PI*R*L) ! Surface area of cylinder in R4
420 PRINT #1,13;STR$(PI*L*R^2) ! Volume of cylinder in R5
430 !
440 ! Now finish up
450 GOTO 'ENDING'
460 !
470 ! Errors come here
480 'FILERR': ! Handle a file error here
490 DISP "ERR:Invalid File Type" @ BEEP 1400 @ WAIT 2
500 ! Null out registers R0 - R5 so they don't get changed.  Ending will
510 ! null out the rest.
520 !
530 PRINT #1,8;"" @ PRINT #1,9;"" @ PRINT #1,10;""
540 PRINT #1,11;"" @ PRINT #1,12;"" @ PRINT #1,13;""
550 !
560 ! We end here
570 'ENDING':
580 !
590 ! Be sure to Null out unused registers
600 PRINT #1,3;"" ! Null LAST X
610 PRINT #1,4;"" @ PRINT #1,5;"" @ PRINT #1,6;"" @ PRINT #1,7;""
620 PRINT #1,14;"" @ PRINT #1,15;"" @ PRINT #1,16;"" @ PRINT #1,17;""
630 !
640 OFF ERROR
650 ASSIGN #1 TO *
660 END
```

Now re-enter the calculator by pressing **C71 [END LINE]**.

Keystrokes	Display		
10[END LINE]	**10.0000**	:F	The length of the cylinder.
20	**20**	:F	Radius of the circle.
[BX]	**:**		Prompt.
RUN TESTFP	**:RUN TESTFP**		BASIC command to run the file TESTFP
	20.0000	:F	Program completed.
[RCL]0	**125.6637**	:F	Circumference.
[RCL]1	**1256.6371**	:F	Area of the circle.
[RCL]2	**5026.5482**	:F	Surface area of the sphere.
[RCL]3	**33510.3216**	:F	Volume of the sphere.
[RCL]4	**1256.6371**	:F	Surface area of the cylinder.
[RCL]5	**12566.3706**	:F	Volume of the cylinder.

[BX] in Computer Scientist Mode

The numbers are converted into string equivalent numbers using the current Base and Word-Size. It is up to the BASIC program to convert the strings into a usable form. It is the responsibility of the user's BASIC program to reformat the numbers into strings in the proper base before writing them back to the C71FILE. The calculator will convert the strings back into internal numbers and put them into the registers according to the current Base and with a Word-Size of 64--no truncation occurs except on the stack which is truncated to Word-Size. The strings must have no leading nulls.

The calculator will not change Word-Size or Base from what it was when it wrote the C71FILE. If the numbers are not in the proper format a conversion error may occur and that register will be not be changed. For example, if the number has HEX digits 8 through F when the calculator is in Octal, then the register will not be changed. However, if the number is in HEX but no digits occur above 7, then the number will be converted, but will have a different value than the one intended. If the number is in Octal and the calculator is in HEX, then the number will be converted and will have a different value than the one intended.

> **Note:** All numbers are converted in unsigned mode and will be read as unsigned. For example, if you are in decimal mode , 2's complement mode, and base 16 mode, a displayed value of -7 will be converted as 65529 in the C71FILE. If your BASIC application puts a minus number like -7 in the file, that line will be treated as a **DATA TYPE** error, will not be converted, and will not change the corresponding register.

Computer Scientist Example

This example gets the address of the FORTHSYS file, its size in bytes, and the address of the next file in the chain by using the BASIC program TESTCS. Then we can check to see if there are any "holes" between reported addresses and sizes.

> **Note:** The FORTHSYS file is always the first file in main memory. The second file cannot be the "workfile". If it is, rename it to something else by first pressing **EDIT [END LINE]**, then pressing **RENAME TO** 'filename' **[END LINE]**.

This program also returns the amount of free memory in HEX bytes. Remember that addresses are in nibbles, while sizes are in bytes. The values are returned in the following registers:

R6 -- Address of FORTHSYS.
R7 -- Size of FORTHSYS.
R8 -- Address of the next file.
R9 -- Amount of free memory.

Enter the following program as **TESTCS** into your HP-71.

```
100 ! The TESTCS program to illustrate BX in Computer Scientist mode
110 !
120 DIM A$[64],A1$
130 REAL A
140 ON ERROR GOTO 'FILERR'
150 !
160 ! Open the file
170 ASSIGN #1 TO C71FILE
180 !
190 ! Check for valid HP-AF format
200 READ #1,0;A$
210 IF A$#"HPAFS" THEN GOTO 'FILERR'
220 READ #1,1;A
230 IF A#15 THEN GOTO 'FILERR'
240 READ #1,2;A
250 IF A#18 THEN GOTO 'FILERR'
260 READ #1,18;A$
270 IF A$#"CS" THEN GOTO 'FILERR'
280 READ #1;A ! Get word size
290 IF A<20 THEN GOTO 'FORMERR' ! Word size is not big enough
300 READ #1;A ! Get base
310 IF A#16 THEN GOTO 'FORMERR' ! Base is not Hex
320 !
330 ! Now get the values
340 PRINT #1,14;ADDR$("FORTHSYS") ! Address of FORTHSYS in R6
350 A$=CAT$(1) ! FORTHSYS is always the first file
360 A1$=A$[18,22] ! Get size field
370 A=VAL(A1$) ! Get the value as a decimal number
380 A1$=DTH$(A) ! Turn it into a HEX string
390 PRINT #1,15;A1$ ! Put size in to R7
400 A$=CAT$(2)[1,8] ! Get name of second file
410 IF A$="workfile" THEN GOTO 'FILERR2' ! workfile cannot be file #2
420 PRINT #1,16;ADDR$(A$) ! Put address of second file in R8
430 PRINT #1,17;DTH$(MEM) ! Put memory size in R9
440 !
450 ! Now finish up
460 GOTO 'ENDING'
470 !
480 ! Errors come here
490 'FILERR': ! Handle a file error here
500 DISP "ERR:Invalid File Type" @ BEEP 1400 @ WAIT 2
510 GOTO 'NULLREG'
520 'FILERR2': ! Handle workfile here
```

```
530 DISP "ERR:workfile not #2" @ BEEP 1400 @ WAIT 2
540 GOTO 'NULLREG'
550 'FORMERR': ! Handle format errors here
560 DISP "Base or WSize wrong" @ BEEP 1400 @ WAIT 2
570 !
580 ! Be sure to Null out registers R6 - R9 so they won't get changed.
590 ! The rest will be cleaned up in Ending
600 'NULLREG':
610 PRINT #1,14;"" @ PRINT #1,15;"" @ PRINT #1,16;"" @ PRINT #1,17;""
620 !
630 ! We end here
640 'ENDING':
650 ! Be sure to Null out unused registers.
660 PRINT #1,3;"" ! Null out LAST X.
670 PRINT #1,4;"" @ PRINT #1,5;"" @ PRINT #1,6;"" @ PRINT #1,7;""
680 PRINT #1,8;"" @ PRINT #1,9;"" @ PRINT #1,10;"" @ PRINT #1,11;""
690 PRINT #1,12;"" @ PRINT #1,13;""
700 OFF ERROR
710 ASSIGN #1 TO *
720 END
```

Now re-enter the calculator by pressing **C71 [END LINE]**.

> **Note:** Your numbers may not exactly match these as they are partially dependent on the amount of memory and other files present in your HP-71.

Keystrokes	**Display**	
[RCL]6	**2FA79 :H**	Address of FORTHSYS.
[RCL]7	**944 :H**	Size in bytes.
2[*]	**892 :H**	Size in nibbles
[+]	**3030B :H**	Ending address of FORTHSYS.
[RCL]8	**3032F :H**	Address of next file.
[X<>Y][-]	**24 :H**	Number of nibbles "unaccounted" for.
[RCL]9	**5A83 :H**	Free memory in main.
[DEC]	**23171 :H**	Free memory in decimal bytes. Remember C71FILE was present

Radians/Degrees Mode

Radians/Degrees mode is maintained separately in the calculator from the BASIC system. This means that if you need Radians in your Calculator but Degrees in your BASIC system they will be maintained that way. However, BX and BFP set the BASIC system to the calculator's Radians/Degrees setting for the duration of the BASIC interface operation. Upon actually exiting the Calculator with [g][EXIT], the proper mode will be restored. This allows Radians/Degrees numbers to be used by any BASIC programs without the user having to worry about Radians/Degrees number compatibility.

Using Storage Registers with BX

Because the conversion routines can possibly convert a number stored in the opposite mode, you should establish some storage register convention. Remember a value that is accidentally converted, will be restored to the register in that converted state. The original number will be lost.

Your program should clear any registers that it does not specifically use. For example

PRINT #1, 15; ""

will prevent register R6 from being altered.

If you do use the storage registers for storing both floating point and Computer Scientist numbers, establish a convention such as: R0 though R5 are used for floating point numbers and R6 through R9 are used for Computer Scientist numbers. Then be sure to set the registers for the other mode to null strings before your program ends.

Handling Errors

If an error occurs while the system is in BASIC from either [BX] or [BFP], there are two ways it can be handled. You can set an **ON ERROR** trap in your program or user-defined function and the error will be handled by the BASIC system and your program. After the program or user-defined function ends, control will be returned normally to the calculator.

If the error is not trapped within BASIC, then the calculator will trap the error, display the error message, and continue operation from within the calculator.

As mentioned above, the C71FILE file will not be read or purged if the calculator handles the error. If BASIC handles the error then the C71FILE file will be read and purged.

User-defined Functions

There are two things that must be avoided in user-defined functions called from [BFP] or a **Memory Lost** will result. The first is copying or purging LEX files. This causes a reconfiguration of the HP-71. The calculator may appear to recover OK after an INIT1; however, when you turn it off or do some other functions the HP-71 will cold start.

The other condition that must be avoided is interrupting a user-defined function with the ATTN key, when that function was called from [BFP]. This also trashes the Soft-FORTH pointers leading to an eventual **Memory Lost**. Again the **Memory Lost** may not happen until you turn the machine off or execute some other function.

Refer to Appendix D, HP-71 System Usage, for additional information regarding the operation of Soft-FORTH.

BASIC Interface Cautions

The calculator sets up its own environment to shield the user from potential system errors and from inadvertently performing the wrong function. The philosophy used in the calculator is that of the Hewlett-Packard series 10 calculators--it is very rigid in what it will and will not allow the user to do. This philosophy is great for doing calculator type functions. However, for the advanced user who wants to program and do special calculations, this rigidity can be a frustrating limitation. Likewise, the freedom to do "whatever" is filled with potential disasters for general use.

The above discussion is provided to help you realize that the calculator is very safe. HOWEVER, when you go into the basic interface functions the calculator does not and cannot protect you. You can use any functions or LEX file enhancements in the BASIC system to do whatever you'd like. The calculator does try to protect itself upon re-entry, but this is not and cannot be an exhaustive check. So the calculator might be in a different state than it was before entry into the BASIC interface. The calculator does save most of its environment before exiting into the BASIC interface.

The most dangerous of all events is the accidental corrupting (ie. having data written that isn't suppose to be there) of the C71RAM or FORTHSYS files. If either one is corrupted, it is possible that the HP-71 will have a MEMORY LOST. The FORTH operating system and calculator system pointers are stored in these two files and thus the calculator could get pointed off into unknown memory upon re-entry.

If the FORTHSYS file is purged, the ROMFORTH lex file will not know that the calculator was active and thus will not cause a re-entry into the calculator upon completion of [BFP] or [BX]. The system will be back in BASIC. However, the next time the calculator is entered by typing C71, the system will behave as if you had pressed [g][EXIT]. The calculator may, however, be slightly different, such as the contents of the X,Y,Z,T stack may be lost. Refer to Radians/Degrees Mode above for information on the potential change in the BASIC system's radians/degrees setting.

Any function that creates files or enlarges files has the potential for causing **Insufficient Memory** errors. The HP-71 handles memory errors differently than other errors. To try to insure that the calculator recovers after a memory error, a second copy of the FORTH system pointers is kept in the C71RAM. The first set is kept in the FORTHSYS file. This second set is used to try to restore the calculator to its operating condition after an **Insufficient Memory** error has occurred.

The problem of **Insufficient Memory** errors has been tested extensively. The calculator recovers properly for all normal causes, such as expanding files, creating new ones, etc. However, it is not possible to test for the effects of the way some LEX files could move and change memory. It is these cases that could cause the above concerns.

The other problem relating to **Insufficient Memory** errors has to do with the integrity of the C71RAM. If that file is purged prior to the occurrence of an **Insufficient Memory** error, the second set of pointers will be gone and the calculator will not be able to recover. This inevitably leads to a **Memory Lost** condition. In fact the purging of C71RAM whenever you are in the BASIC system through [BFP] or [BX] is likely to cause a **Memory Lost** condition.

If the calculator does recover after an **Insufficient Memory** error when C71RAM has been purged or corrupted, the calculator will probably not be performing normally. Because most of the calculator's pointers were destroyed, the system may have written in some arbitrary part of memory. So you should check all RAM or IRAM files for integrity. The best solution may be the INIT 3 and cleaning out the IRAMs and restarting the system.

If while in the BASIC system, the calculator times out or it is shutoff using the f[ON] key sequence while at the command line colon prompt, the system will stay in the BASIC system when next turned on. A programmatic BYE or typing BYE on the command line will turn the system off, however it will re-enter the calculator when next turned on.

If the system does get left in the BASIC system, the calculator will have restored the BASIC system to it previous state with the exception of the Degrees/Radians flag. This flag will be in the calculator's state. Likewise the calculator status was properly saved, so the calculator will be able to be re-entered and function normally by typing C71.

While the items above sound serious--and they are--, by exercising some good programming practice and by being careful to insure the integrity of the FORTHSYS and C71RAM files, the normal error handling of the calculator should prevent any problems even in **Insufficient Memory** conditions.

Appendix A

Error Messages

Can't solve

Functions: [i] and [IRR].

Meaning: Sixteen iterations of the numerical methods solution have been done, and it has not converged close enough for an answer to be given. You might try putting a value in *i* that is close to what you think the answer is. The calculations start with *i*, so if *i* is close, it might converge. However, if it didn't converge after sixteen iterations using the built in seed, it likely is not solvable.

Cash flows are 0

Functions: [i] and [IRR].

Meaning: Cannot calculate an interest because all of the N_j's times their respective CF_j's are 0. Thus all of the cash flows are 0. [i] sets up as if calculating an IRR, then it uses the IRR routines to find *i*. For *i* it means that PMT, PV, and FV are all 0.

CFj array is full

Function: [CFj].

Meaning: The calculator can use only 20 different cash flows (grouped or ungrouped) and an attempt was just made to enter the 21st.

Data Type

Functions: Applies to all Floating Point Functions.

Meaning: The IEEE defaults are set to off and a number larger or smaller than the range of the computer was entered. In [BFP] the prompt line cannot be interpreted.

While in Engineering/Business, recalled a Computer Scientist number that cannot be converted into a floating point format number.

Invalid Argument

Function: [AMORT].

Meaning: The number of periods to amortize (in the X-register) is either negative or has a fractional part. Amortization must have a positive or 0 integer count.

Function: [C^1/N].

Meaning:
1. The number of roots or the root number (that is, the value in either the X- or Y-registers) is negative.
2. The number of roots is 0.
3. The root number you want is greater than the number of roots.

Function: [C/].

Meaning: An attempt was made to do a complex division by 0.

Function: [HYP][SIN], [HYP][COS], [HYP][TAN], [AHYP][SIN], [AHYP][COS], [AHYP][TAN], [->P], [->R], [->H], [->H.MS], [H.MS+], [H.MS-], [%], [Δ%], [%T], [ΔDYS], [C+], [C-], [C*], [C/], [1/Z], [SINZ], [COSZ], [TANZ], [e^z], [LN Z], [Z^w], [a^z], [$LOG_a Z$], [$Z^{1/w}$], [$Z^{1/n}$], [Z^n], [|z|].

Meaning: One of the values was not valid for the specified calculation.

Invalid Date

Functions: [ΔDYS] and [DATE].

Meaning: The date entered in either the X- or the Y-registers is not a proper date.

Invalid File Type

Function: [BX].

Meaning: When BX went out to C71FILE to read in the register values, the HP-AF format was not correct, so it did not know how to read the file. The file will not be read, and no register values will be changed.

Invalid index; N = -1

Functions: [STO][Nj], [RCL][Nj], [STO][CFj] and [RCL][CFj].

Meaning: The number in the n register has been decremented to -1 which is not a valid index for the cash flow array.

LOG(neg)

Functions: [LN] and [LOG]. [LOGP1] for x less than -1.

Meaning: An attempt was made to take the logarithm of a negative number.

N is too large

Functions: [STO][Nj], [RCL][Nj], [STO][CFj], [RCL][CFj].

Meaning: The number in the n register has been incremented to 21 which is not a valid index for the cash flow array.

N must be an integer

Functions: [IRR] and [NPV].

Meaning: IRR and NPV cannot use an N with a fractional part when doing IRR or NPV calculations.

Functions: [STO][Nj], [RCL][Nj], [STO][CFj], and [RCL][CFj].

Meaning: That the number in the n register must be an integer for doing valid stores and recalls from the cash flow array.

N must be > 0

Functions: [PMT], [i], [IRR].

Meaning: These financial calculations cannot use N when it is 0 or negative. For these functions N must be positive.

N must be > or = 0

Function: [NPV].

Meaning: NPV cannot be calculated using a negative N. However, it can use 0 where IRR cannot use 0.

Neg^Non-int

Function: [Y^x].

Meaning: An attempt was made to raise a negative number in the Y-register to a noninteger number in the X-register

No positive solution

Functions: [i] and [IRR].

Meaning: The sum of the respective N_j * CF_j did not change sign once. This means *i* can only be negative. The financial calculations only understand positive *i*.

No solution...

Function: [n].

Meaning: Cannot calculate N.

Function: [PMT].

Meaning: Cannot calculate a valid PMT because N * i% is negative and PV and FV are both 0.

Not a flag...

Functions: [SF], [CF], and [F?].

Meaning: The number in X is not a valid flag number.

Not positive...

Function: [N].

Meaning: The calculated N is not positive. The routines for finance only understand a positive N.

SQR(neg)

Function: [√x].

Meaning: An attempt was made to take the square root of a negative number.

Too large...

Functions: [WSIZE], [SB], [CB], [B?], [RLn], [RRn], [RLCn], [RRCn], [MASKL], and [MASKR].

Meaning: The number in X is too large for the specified operation.

/ZERO

Functions: [/] and [RMD] in Computer Scientist mode.
[/] in Engineering/Business mode while IEEE traps are set to default off.

Meaning: An attempt was made to divide by zero while in the Computer Scientist mode. There are no overflow or underflow conditions in this mode.
An attempt to divide a number by zero while in Engineering/Business mode with IEEE traps set to off, so no automatic over flow occurs.

0/0

Function: [/].

Meaning: An attempt was made to divide a zero in the Y-register by a zero in the X-register.

If the IEEE traps are set to anything other than **DEFAULT ON** then any of the HP-71 math errors (error numbers 1 through 21) could be displayed. Refer to the HP-71 Owner's Manual or the HP-71 Reference Manual for additional information.

Appendix B

Stack Lift, Last X, and Flags

Your C71 calculator has been designed to operate in a natural manner. As you have seen as you worked through this handbook, you are seldom required to think about the operation or the automatic memory stack--you merely work through calculations in the same way you would with a pencil and paper, performing one operation at a time.

There may be occasions, however, when you wish to know the effect of a particular operation upon the stack. The following explanation should help you.

Digit Entry Termination

Most operations on the calculator terminate digit entry. This means that the calculator knows that any digits you key in after any of these operations are part of a new number. (The [CHS]--except in Computer Scientist mode, [.], [EEX], and [<-] operations do not terminate digit entry.)

Stack Lift

There are three types of operations on the calculator, depending upon how they affect the stack lift. These are stack-*disabling* operations, stack-*enabling* operations, and *neutral* operations.

Disabling Operations

There are four stack-disabling operations on the calculator. These operations *disable* the stack lift, so that a number keyed in after one of these disabling operations writes over the current number in the displayed X-register and the stack does not lift. These special disabling operations are:

[END LINE] [CLx] [Σ+] [Σ-]

Enabling Operations

Most of the operations on the keyboard, including one- and two-number mathematical functions like [x^2] and [*], are stack enabling operations. These operations *enable* the stack lift, so that a number keyed in after one of the enabling operations lifts the stack. Note that switching from Computer Scientist to Engineering/Business Mode, from Engineering/Business mode to Computer Scientist mode, turning the calculator off, then on, and entering the calculator from BASIC are enabling operations. Additionally, all errors enable the stack.

T ->				
Z ->				
Y ->		**4.0000**	**4.0000**	**53.1301**
X ->	**4**	**4.0000**	**3**	**5.0000**
Keys ->	4	[END LINE] Stack disabled.	3 No stack lift.	[g][->P] Stack enabled.

```
  T ->
  Z ->
  Y ->   53.1301      4.0000        4.0000       53.1301
  X ->    0.0000        7           1.0000          9

Keys -> [g][CLx]        7            [Σ+]           9
         Stack      No stack        Stack      No stack
       disabled.      lift.       disabled       lift.
```

Neutral Operations

Some operations, like [CHS] (except as noted below) and [FIX] are neutral; that is, they do not alter the previous status of the stack lift. Thus, if you disable the stack lift by pressing [END LINE], then press [f][FIX] *n* and key in a new number, that number will write over the number in the X-register and the stack will not lift. Similarly, if you have previously enabled the stack lift by executing, say [X^2], then executed a [FIX] instruction followed by a digit entry sequence, the stack will lift.

The following are neutral operations:

[FIX]
[ENG]
[SCI]
[STD]
[DEG]
[RAD]
[CLR FIN]
[CLR REG]
[CLR STAT]
I[BEG]
[END]
[D.MY]
[M.DY]

[HEX]
[DEC]
[OCT]
[BIN]
[WINDOW]
[SCRL]
[SCRR]
[STATUS]
Set Complement [1's] [2's] [Uns]

[CHS] has the behaviors defined below.

In Engineering/Business:
 * If digit entry has not been terminated, it is neutral.
 * If digit entry has been terminated, it enables stack lift.

In Computer Scientist:
 * It terminates key entry and enables stack lift.

LAST X

The following operations save x in LAST X:

[-]	[Σ+]	[e^x]	[$\sqrt{x}$]	[RND]
[+]	[Σ-]	[LOG]	[x^2]	[HYP][SIN]
[*]	[%]	[10^x]	[1/x]	[HYP][COS]
[/]	[Δ%]	[SIN]	[y^x]	[HYP][TAN]
[->H.MS]	[%T]	[ASIN]	[->R]	[AHYP][SIN]
[->H]	[x!]	[COS]	[->P]	[AHYP][COS]
[H.MS+]	[FP]	[ACOS]	[->RAD]	[AHYP][TAN]
[H.MS-]	[IP]	[TAN]	[->DEG]	[ΔDYS]
[ABS]	[LN]	[ATAN]	[BFP]	[DATE]
[TOTAL]	[C+]	[SIN Z]	[e^z]	[LN Z]
[CORR]	[C-]	[COS Z]	[a^z]	[LOGa Z]
[MEAN]	[C*]	[TAN Z]	[Z^n]	[Z^w]
[SD]	[C/]	[1/z]	[Z^1/n]	[Z^1/w]
[P.VAL]	[|z|]			
[LR]				

[XOR]	[LJ]	[RLC]	[RL]	[SB]
[NOT]	[ASR]	[RRC]	[RR]	[CB]
[OR]	[SL]	[RLCn]	[RLn]	[B?]
[AND]	[SR]	[RRCn]	[RRn]	[#B]
[RMD]	[WSIZE]	[MASKL]	[MASKR]	

[CHS] in Computer Scientist mode.

CS Functions That Affect CS Flags

Two important flags, the carry flag (3) and the out-of-range flag (4), are affected (set or cleared) bye certain arithmetic and shifting functions in *Computer Scientist mode.* These functions are listed below.

X = sets or clears   - = no effect

Function	Effect On Carry(3)	Out-Of-Range (4)	Registers Used: Operand(s)	Result
[+]	X	X	X,Y	X
[-]	X	X	X,Y	X
[*]	-	X	X,Y	X
[/]	X	X	X,Y	X
[ABS]	-	X	X	X
[CHS]	-	X	X	X
[SL]	X	-	X	X
[SR]	X	-	X	X
[ASR]	X	-	X	X
[RL]	X	-	X	X
[RR]	X	-	X	X
[RLC]	X	-	X	X
[RRC]	X	-	X	X
[RLn]	X	-	X,Y	X
[RRn]	X	-	X,Y	X
[RLCn]	X	-	X,Y	X
[RRCn]	X	-	X,Y	X

Appendix C
Owner's Information

Maintenance

The HP-71 C71 Calculator module does not require maintenance. However, there are several precautions listed below that you should observe.

CAUTION

* Do not place fingers, tools, or other objects into the plug in ports. Damage to plug-in module contacts and the computer's internal circuitry may result.

* Turn off the computer (press [f][ON]) before installing or removing a plug-in module.

* If a module jams when inserted into a port, it may be upside down. Attempting to force it further may result in damage to the computer or the module.

* Handle the plug-in modules very carefully while they are out of the computer. Do not insert any objects in the module connector socket. Always keep a blank module in the computer's port when a module is not installed. Failure to observe these cautions may result in damage to the module or to the computer.

Limited 90-Day Warranty

What We Will Do

The C71 Calculator module is warranted by Corvallis Micro Technology against defects in materials and workmanship affecting electronic and mechanical performance, but not software content, for 90 days from the date of original purchase. If you sell your unit or give it as a gift, the warranty is transferred to the new owner and remains in effect for the original 90-day period. During the warranty period, we will repair or, at our option, replace at no charge a product that proves to be defective, provided you return the product, shipping prepaid, to Corvallis Micro Technology.

What Is Not Covered

The warranty does not apply if the product has been damaged by accident or misuse or as the result of service or modification by other than Corvallis Micro Technology.

No other express warranty is given. The repair or replacement of a product is your exclusive remedy. ANY OTHER IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS IS LIMITED TO THE 90-DAY DURATION OF THIS WRITTEN WARRANTY. Some states, provinces, or countries do not allow limitation on how long an implied warranty lasts, so the above limitation may not apply to you. IN NO EVENT SHALL CORVALLIS MICRO TECHNOLOGY COMPANY BE LIABLE FOR CONSEQUENTIAL DAMAGES. Some states, provinces, or countries do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Warranty for Consumer Transactions in the United Kingdom

This warranty shall not apply to consumer transactions and shall not affect the statutory rights of a consumer. In relation to such transactions, the rights and obligations of Seller and Buyer shall be determined by statute.

Obligation to Make Changes

Products are sold on the basis of specifications applicable at the time of manufacture. Corvallis Micro Technology shall have no obligation to modify or update products once sold.

Warranty Information

If you have any questions concerning this warranty, please contact Corvallis Micro Technology at:

Corvallis Micro Technology
895 N.W. Grand Ave.
Corvallis, OR 97330
Telephone: (503) 752-5456

Service

Corvallis Micro Technology maintains its own service center. You may have your unit repaired at Corvallis Micro Technology any time it needs service, whether the unit is under warranty or not. There is a charge for repairs after the 90-day warranty period.

Corvallis Micro Technology products are normally repaired and reshipped within 5 working days of receipt. This is an average time and could vary depending upon the time of year and the work load at the service center. The total time you are without your unit will depend largely on the shipping time.

Obtaining Repair Service

The Corvallis Micro Technology Service Center is located in Corvallis, Oregon:

Corvallis Micro Technology
Service Department
895 N.W. Grand Ave.
Corvallis, Oregon 97330, U.S.A.

Telephone: (503) 752-5456

Service Repair Charge

There is a standard repair charge for out-of-warranty repairs. The repair charges include all labor and materials. In the United States, the full charge is subject to the customer's local sales tax.

Products damaged by accident or misuse are not covered by the fixed repair charges. In these situations, repair charges will be individually determined based on time and materials.

Service Warranty

Any out-of-warranty repairs are warranted against defects in materials and workmanship for a period of 90 days from date of service.

Shipping Instructions

Should your unit require service, return it with the following items:

* A completed Service Card, including a description of the problem and system setup when the problem occurred.

* A sales receipt or other documentary proof of purchase date if the one-year warranty has not expired.

The product, the Service Card, a brief description of the problem and system configuration, and (if required) the proof of purchase date should be packaged in the original shipping case or other adequate protective packaging to prevent in-transit damage. Such damage is not covered by the 90-day limited warranty; Corvallis Micro Technology suggests that you insure the shipment to the service center. The packaged unit should be shipped to Corvallis Micro Technology.

Whether the unit is under warranty or not, it is your responsibility to pay shipping charges for delivery to Corvallis Micro Technology.

After warranty repairs are completed, the service center returns the unit with postage prepaid. On out-of-warranty repairs in the United States and some other countries, the unit is returned C.O.D. (covering shipping costs and the service charge).

Further Information

Service contracts are not available. Circuitry and designs are proprietary to Corvallis Micro Technology, and service manuals are not available to customers.

Technical Assistance

The keystroke procedures and program material in this manual are supplied with the assumption that the user has a working knowledge of the concepts and terminology used. Corvallis Micro Technology's technical support is limited to explanations of operating procedures used in the manual and verification of answers given in the examples. Should you need further assistance, you may write to:

Corvallis Micro Technology
Customer Service
895 N.W. Grand Ave
Corvallis, OR 97330

Dealer and Product Information

For dealer locations, product information, and prices, please call (503) 752-5456.

Appendix D
HP-71 System Usage

This section details the amount of memory required, file names used, variables used, and other technical aspects of C71's interaction with the HP-71.

File Names and Memory Usage

The following files are created and used by C71:

File	Type	Size	Used For:
FORTHSYS	FORTH	1102	Soft-FORTH system RAM file.
C71RAM	FORTH	640	C71 Calculator RAM file.
C71FILE	DATA	1292	Data file for register exchange with BASIC files from the function [BX].

If any files exist in your HP-71 with those names prior to running C71, it will cause unexpected operations. If another file with the name of either of the FORTH files (FORTHSYS or C71RAM) exist, the calculator will give an **Invalid File Type** error message. If a file named C71FILE exists prior to pressing [BX], a **File Exists** error message will be displayed.

The HP-71 must have enough memory to accommodate FORTHSYS and C71RAM before the calculator can be entered. C71 creates these two files when you first press **C71** [END LINE]. If you never use the [BX] function, then you do not need memory for C71FILE. C71FILE is created when [BX] is pressed, and purged by the calculator when the function completes.

As there is some system overhead required to create files, your HP-71 will need a few bytes more than the actual size of the files.

Variables Used by C71

C71 uses several HP-71 BASIC variables in conjunction with the statistics functions and with [BX]. The following is a list of the variables and what they are used for.

Statistics:

R7(2)	The statistics array variable, declared using the BASIC **STAT** command.
R8	Used to get the slope from the BASIC **LR** function.
R9	Used to get the intercept from the BASIC **LR** function.

[BX]:

R9$[68]	Used to read the string representations of registers from the C71FILE.
R9	Used to read certain HP-AF file format information from the C71FILE.

Additional Technical Information

Overview

C71 is based on an interaction of three systems. First, it uses the HP-71 operating system to control all file handling, number control for floating point numbers, statistics, and many of the mathematical operations. Second, it uses an alternative operating system called Soft-FORTH, which is designed to co-exist with the HP-71 native BASIC operating system. Soft-FORTH is a special ROM version of the HP-71 FORTH/Assembler ROM. This version is designed to be put into application ROMs, and allows multiple Soft-FORTH applications to be present in the HP-71. Third, the actual calculator--C71--is written in a combination of FORTH and assembly code (FORTH primitives).

The Soft-FORTH uses the FORTHSYS file as its allocated, dedicated RAM area. While C71 uses C71RAM as its allocated, dedicated RAM area. If multiple Soft-FORTH applications are present in your HP-71, they will all use FORTHSYS while they are running. However, only C71 uses C71RAM. Your other Soft-FORTH applications will have their own dedicated RAM areas.

FORTHSYS maintains the FORTH stacks, pointers, and the X-, Y-, Z-, T-, and LAST X registers.

C71RAM maintains the storage registers, financial registers, internally-used working registers, and some special mainframe-save variables. In order to operate properly, C71 must over-write certain mainframe operating status information. This status information is saved in C71RAM upon entry into the calculator, and restored upon exit. Additional information on what is saved and potential problems are discussed below.

Display Delay

Upon entry, C71 stores the mainframe's delay setting and then sets the delay to **DELAY 0,0**. When a function is performed that goes into the basic interpreter, the machine's original delay setting is restored. If anything is displayed while in BASIC, the display delay and scrolling set in the BASIC system will be used. The two functions that provide an interface to BASIC are [BX] and [BFP].

Statistics and BASIC Variables

While in the BASIC system's main environment, whether you're in [BFP], [BX], or exited from the calculator, if you do a DESTROY ALL or DESTROY R7, you will have destroyed your statistics array and will get an **Invalid stat array** error message next time you press a statistic function key, like [Σ+]. You solve this by pressing the [CLR STAT] key. The statistics will then work just fine. The only statistic function this does not apply to is the [CLR STAT] function.

R7 needs to be preserved to maintain a valid stat array entered with the [Σ+] and [Σ-] keys from within the calculator. R9$, R9, and R8 are temporary variables that are a don't care on entry.

Any previous value will be lost upon using the above functions, so don't rely on them outside of the calculator.

Purging System Files

Purging the system files while exited from the calculator will cause no problem with the operation of the calculator. However, when the calculator is re-entered, the calculator will be set to its default values.

Purging C71RAM will cause the calculator to re-initialize the stat array the next time the [Σ+] key is pressed. The other stat functions may work OK because the stat array is kept by the mainframe and not in C71RAM. [Σ+] sets a flag in C71RAM when it has created a valid stat array in the mainframe. If C71RAM is purged, the calculator has no way of knowing if there is a valid array or not. So it creates a new array and sets the stat valid flag. CLRSTAT clears the stat valid flag.

A copy of the X-, Y-, Z-, T-, and LAST X register are kept in C71RAM when the calculator is exited. So there should be no change in the previous values or status of the calculator if FORTHSYS is purged. FORTHSYS is used only during the actual running of the calculator and not for value storage.

If C71RAM is purged while in **[BX]** or **[BFP]**, the HP-71 will perform a **Memory Lost**. The Soft-FORTH system is not able to recreate the file and begin running again. The flags the tell the Soft-FORTH system that it is active and in the middle of the **[BX]** or **[BFP]** routines (the FORTH pointers) are maintained in the FORTHSYS file. However, all of the calculator's status information is maintained in C71RAM, where it was saved prior to executing **[BX]** or **[BFP]**. So the Soft-FORTH system has no way of recovering the proper operation of the calculator as it tries to reestablish its pointers and begin running again. It gets pointed off into some unknown part of memory and ends up causing a **Memory Lost**.

If FORTHSYS is purged while in **[BX]** or **[BFP]**, the Soft-FORTH system will have no way of knowing that it was active. When the mainframe polls to reenter Soft-FORTH, Soft-FORTH will just return leaving the system in BASIC, otherwise undamaged. The Radians setting may be different, however, all of the other BASIC system setting will be properly restored. Refer to the section on the BASIC interface for additional information on the Radians setting.

Display Contrast

The display contrast is constantly being altered. Upon entry into C71, your contrast setting is read and stored. As the calculator operates, you will notice the display being blanked at times. This is done by setting the contrast to 0 and then restoring it. If something really gets fouled up and you can no longer see the display, do an INIT1 to reset the system to get the display back, purge FORTHSYS and C71RAM, then re-set the display contrast to where you like it. INIT1 while in the calculator will reset the system back to BASIC. Some of your other system settings may no longer be as you like them as you were returned to BASIC before C71 could restore your BASIC system. Such things as your delay setting, your IEEE defaults, and your display mode may need to be reset.

C71 System Modes

The calculator maintains a separate display mode (fix, sci,...) and a separate radians-degrees mode from that of the BASIC system. If you are set to radians mode and fix 2 in the calculator, your BASIC system can be set to degrees and STD. The calculator will keep these separated by saving and restoring them at the proper times.

The one mode that is most likely to get mixed is the radians-degrees mode as the calculator's radians/degrees setting is maintained for [BX] and [BFP] calculations. This is done so you won't have to convert units. Every effort is used to maintain the separate status of these when entering and exiting the calculator. However, if an error occurs such that the calculator cannot recover and you're left in BASIC, then the calculator's mode will be in effect in your BASIC system.

Mainframe State Usage

The calculator tries to use your machine states as much as is reasonable. However some need to be altered for the proper running of the calculator, like the delay setting.

The following settings are modified and maintained separately by the calculator:

Function	**Calculator's Setting**
Display Delay	DELAY 0,0
Display Mode	User selectable from within C71
Display Width	WIDTH 80
Radians/Degrees	User selectable from within C71
ATTN key	Disabled

The remainder of the mainframe's states are used by C71. For example, if you have set beep off in BASIC, then it will be off in the calculator.

The state of IEEE math traps and defaults are used by the calculator with the following exception. C71 functions that are calculated using FORTH floating point functions force **DEFAULT ON** and **BEEP OFF**. Thus if overflows or underflows occur you will not be continuously beeped at as they are occurring. Overflows occurring during a calculation will not accurately reflect whether the final answer would be an overflow or an underflow. Likewise when these calculations are over no warning message will be given that an overflow or underflow occurred. Also if the final result is an overflow or an underflow, that result will not be displayed as INFs and EPS's but will be displayed as the largest or smallest number the HP-71 uses in DEFAULT ON mode. The beeper status will be restored as well as your default status when the calculation is finished.

The IEEE math trap settings are only applicable to Engineering/Business mode, Computer Scientist mode does not allow for this type of traps or exceptions.

The calculator makes use of the system and user flags. All of the flags are stored away before entering the calculator and restored when leaving (including BX, BFP). So your basic machine should always be in the state you had it before entering the calculator. However in the unlikely event that a really obscure error dumps you into the basic without restoring the basic state, you may have to do some clean-up before getting your system back where it should be.

BASIC system status' used by the calculator:

Warning suppress (flag -1)
beep off (flag -2)
continuous on (flag -3)
beep loud (flag -25)
IEEE traps
round off setting (flags -11,-12)
random number seed

If flag -1 is clear and the beep is set to ON (flag -2 is clear), a normal warning will not be displayed, but the HP-71 will beep. For example, with **DEFAULT ON**, if you do 2[END LINE]0[/], the calculator will beep and display the overflow result, but will not display the warning message. Whereas, in BASIC **2/0** would display **WRN: /ZERO** for the delay setting, and then display the overflow result. If you key in a number too large for the calculator (1E+600) and press [END LINE], with **DEFAULT ON**, the calculator will beep and display the overflow number, 9.99999999999E499. If you key in a number too large for the calculator (1E+600) and then press a function key that will increase the number (like [X!]), the calculator will beep twice--once as it assigns the overflow number and a second time as [X!] overflows.

The following system status' are preserved, but not used by the calculator:

user defined keys
user flags

The following is a list of the functions that are performed using the FORTH floating point functions and thus, force a **DEFAULT ON** setting:

AMORT	NPV	->H
INT	IRR	->H.MS
n	%	H.MS+
i	Δ%	H.MS-
PV	%T	All the complex number functions
PMT	->P	All the Hyperbolic functions
FV	->R	

C71 Default Settings

When a new C71RAM file and FORTHSYS file are created, the following are the calculator's default status setting when the calculator first starts:

Delay 0,0 (cannot be changed)
Display mode--STD
Engineering/Business mode
End mode (financial)
Degrees mode
Random number seed--same as mainframe's (uses mainframe's routines)
X-, Y-, Z-, T-, LAST X registers are clear
Finance registers are clear
Storage registers are clear
Statistics array -- presumed clear (see note above)
IEEE Math traps--same as mainframe's

16 bit wordsize
HEX base
carry clear (flag 3)
out-of-range clear (flag 4)
Unsigned complement math
Leading zeros not displayed

ATTN Key

The **ATTN** key on the HP-71 is normally enabled to interrupt a running program. If the **ATTN** key were to be pressed while in Soft-FORTH, the Soft-FORTH system would terminate and return you to the BASIC system. The **ATTN** key, however, can be disabled from interrupting a running program. With all of the keystrokes available around the **ATTN** key it was felt that there would be too great of a chance of accidentally pressing it. If it were allowed to terminate the running of the calculator, the calculator would not have a chance to restore the BASIC system environment and to save its own. Thus, when the calculator is reentered, the status of the calculator would also be different.

For the above reasons, the **ATTN** key is disabled during the actual calculator and cannot be used to interrupt the program. If a run-away condition (infinite loop) should occur or the calculator gets into a bad state, an INIT1 will need to be used to recover.

Any time the calculator is returned to the BASIC environment, the **ATTN** key is restored to its program interruption capability state. The functions that restore this state are: **[f][OFF]**, **[g][EXIT]**, **[f][BFP]**, and **[f][BX]**.

Program Deallocation

When the calculator times out or you press **[f][OFF]**, the calculator is turned off with a BASIC programmable **BYE** after restoring the BASIC environment and saving the calculator's environment. When the **[ON]** (ATTN) key is pressed to turn the HP-71 back on, the HP-71 will try to continue running the program that was being run prior to the **BYE**. However, if a plug-in module is put in or removed while the HP-71 is off from a programmable **BYE**, the HP-71 deallocates the program and will not try to continue running the program.

Thus, if you turn the calculator and the HP-71 off with a **[f][OFF]**, and then plug in a module or remove a module, the calculator will not come back on when you press the **ATTN** key. You will be left in BASIC. However, as the BASIC state was restored and the calculator state saved, pressing **C71** will restart the calculator.

If you turn off the HP-71 while at the basic prompt from **[BX]** or **[BFP]** by pressing **f OFF** or allowing the HP-71 to timeout, the program will also deallocate and you'll be left in the BASIC system. A programmable bye in a basic program or a user-defined function does not deallocate the program.

Cautions

Do not copy or purge LEX files from within a user-defined function executed from **[BFP]**. Soft-FORTH cannot recover properly and a **MEMORY LOST** is the final result. Usually, an **INIT1** will set things up for a **System Error** message. Once this message is given an **INIT3** is the only resolution. (Refer to section 12 on the BASIC interface for more detail.)

If your user-defined function is interrupted with the ATTN key and the user defined function was called from [BFP], the Soft-FORTH will not recover and an eventual **MEMORY LOST** message will be displayed. (Refer to section 12 on the BASIC interface for more detail.)

Function Index

Function Index