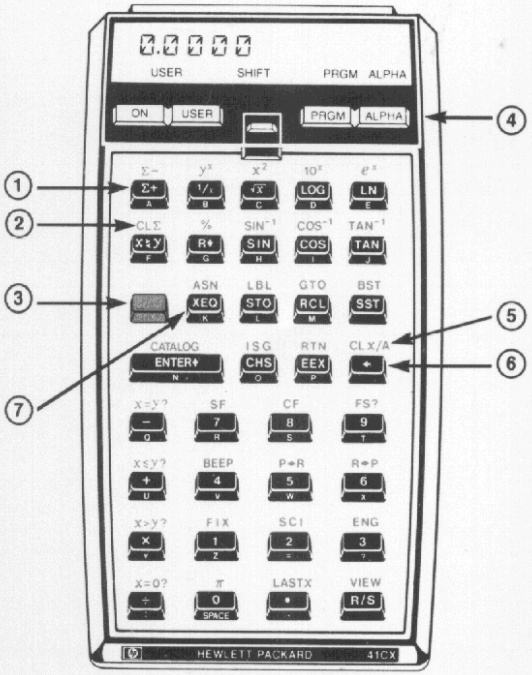
HP-41CX
Quick Reference Guide

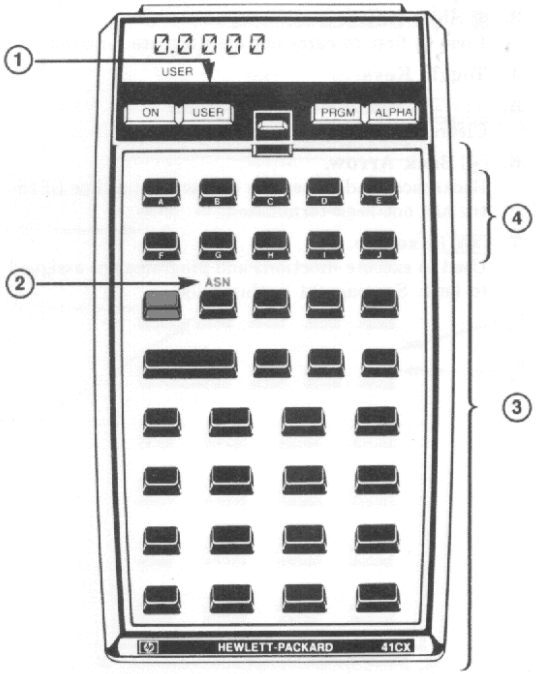
Contents

The Normal Keyboard



1. **Primary Function.**
2. **Alternate Function.**
3. **Shift Key.**
   Press ■ first to carry out an alternate function.
4. **Toggle Keys.**
5. CLX/A **Clear X or Clear Alpha.**
   Clears the entire register.
6. ← **Back Arrow.**
   Backspaces and erases one character at a time (if entry has not been terminated).
7. XEQ **Execute.**
   Used to execute functions and programs not assigned to keys. See page 14 in this guide.

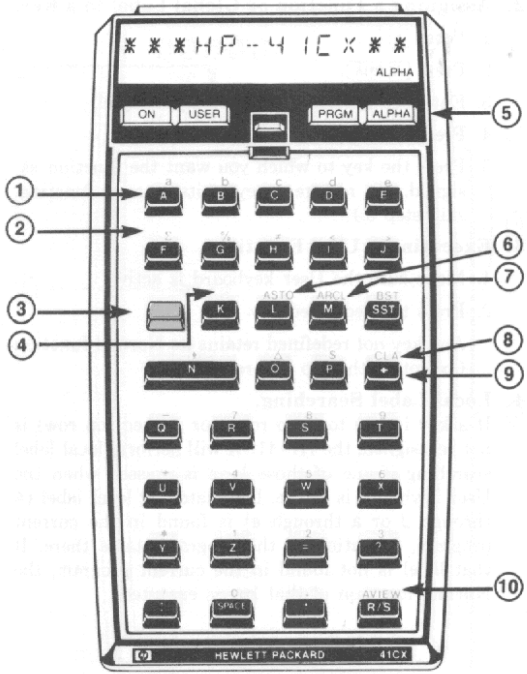
The User Keyboard



1. [USER] **User.**
   Activates and deactivates the User keyboard.
2. **Assigning a Function or Global Label to a Key.**
   1. Press [ASN].
   2. Press [ALPHA].
   3. Enter the function name or global label.
   4. Press [ALPHA].
   5. Press the key to which you want the function assigned. (To restore a key to its Normal function, skip step 3.)
3. **Executing a User Function.**
   1. Make sure the User keyboard is active.
   2. Press the redefined key.

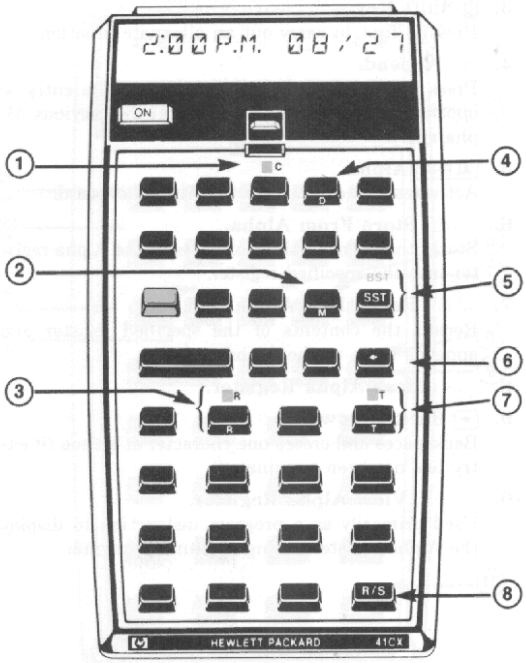
   Any key *not* redefined retains its Normal function (except in the top two rows).
4. **Local Label Searching.**
   If a key in the top two rows (or shifted top row) is not reassigned, the HP-41CX will perform local label searching if one of those keys is pressed when the User keyboard is active. If a matching local label (A through J or a through e) is found in the current program, execution of the program starts there. If that label is not found in the current program, the Normal function of that key is executed.

The Alpha Keyboard



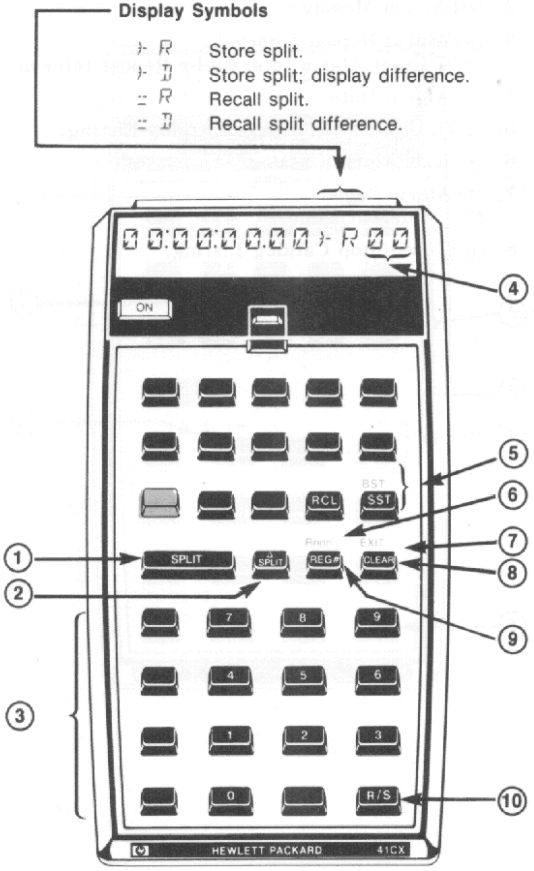
1. **Primary Function.**
2. **Alternate Function.**
3. **Shift Key.**
   Press ■ first to carry out an alternate function.
4. **Append.**
   Press ⊢ first to have the following Alpha entry *be appended to* (rather than overwrite) the previous Alpha entry.
5. **Alpha.**
   Activates and deactivates the Alpha keyboard.
6. **Store From Alpha.**
   Stores the leftmost six characters of the Alpha register into the specified register.
7. **Recall Into Alpha.**
   Recalls the contents of the specified register and appends them to the Alpha register.
8. **Clear Alpha Register.**
9. **Back Arrow.**
   Backspaces and erases one character at a time (if entry has not been terminated).
10. **View Alpha Register.**
    Used primarily as a program instruction to display the Alpha register during a running program.

The Alarm Catalog Keyboard



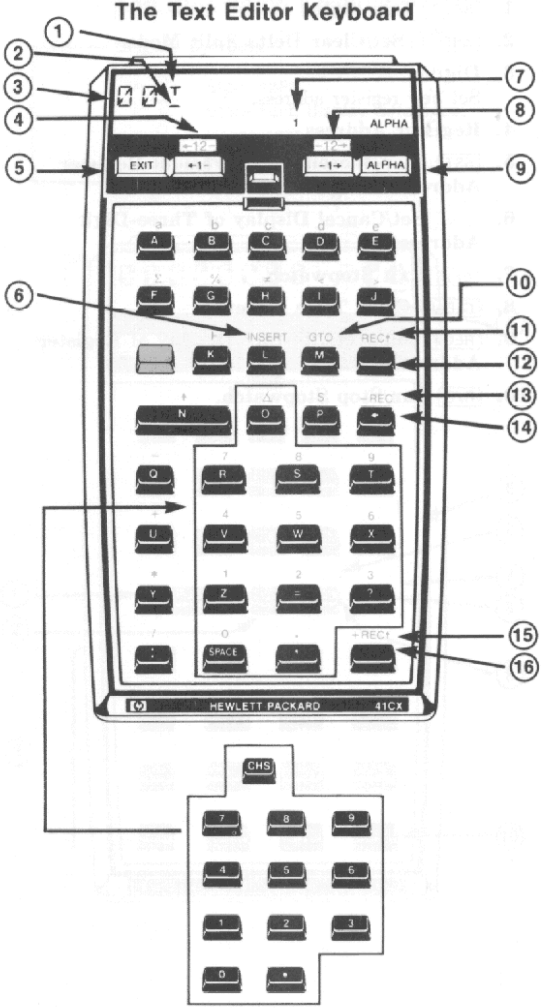
1. [shift] [C] **Clear Alarm.**
2. [M] **Alarm Message.**
3. [R] **Alarm Repeat Interval.**
   [shift] [R] **Reset Alarm Interval by Repeat Interval.**
4. [D] **Alarm Date.**
5. [SST], [BST] **Step Through Catalog Listing.**
6. [←] **Exit Alarm Catalog.**
7. [T] **Alarm Time.**
   [shift] [T] **Current Time.**
8. [R/S] **Run/Stop Catalog Listing.**

The Stopwatch Keyboard



1. [SPLIT] **Take Split.**
2. [ΔSPLIT] **Set/Clear Delta Split Mode.**
3. **Digits.**
   Set new register address.
4. **Register Address.**
5. [SST], [BST] **Increment/Decrement Register Address.**
6. [Rnnn] **Set/Cancel Display of Three-Digit Address.**
7. [EXIT] **Exit Stopwatch.**
8. [CLEAR] **Clear Time to Zero.**
9. [REG#] **Suppress/Restore Display of Register Address.**
10. [R/S] **Run/Stop Stopwatch.**

The Text Editor Keyboard



Numeric Keypad

1. Empty-Record Indicator.
2. Cursor (Pointer).
3. Record Number.
4. ←1, ←12 Move Cursor Left.
5. EXIT Exit Text Editor.
6. INSERT Insert/Replace Mode Toggle.
7. Insert Mode Active.
8. 1→, 12→ Move Cursor Right.
9. ALPHA Alpha/Numeric Keyboard Toggle.
10. GTO Go To Record *nnn*.
11. REC↑ Go To Previous Record.
12. REC↓ Go To Next Record.
13. -REC Delete Record.
14. ← Delete Character.
15. +REC↑ Insert New Record Before Current Record.
16. +REC↓ Insert New Record After Current Record.

How to Execute Functions (Alpha Execution)

If a function has its own key (whether on the Normal keyboard or the User keyboard), you can perform its operation by pressing that key—such as for [1/x]—or by pressing the shift key and then that key—such as for . (Remember to supply any necessary numbers or labels first.)

If a function does not appear on the keyboard—such as [TIME]—you can perform it using either Alpha execution or a User-defined key on the User keyboard. How to assign functions to User keys is shown on page 5 of this guide. Alpha execution is shown below:

1. Press [XEQ].
2. Press [ALPHA] to activate the Alpha keyboard.
3. Spell out the Alpha name of the desired function, or the global label of the desired program.
4. Press [ALPHA] to deactivate the Alpha keyboard and end the procedure.

If the function needs a parameter, it will cue for it with the _ input cue.

Function Set

This is an alphabetical list of the HP-41CX functions, including brief definitions. For a more detailed summary of these functions, refer to the Function Tables in volume 2 of the owner's manual. For page references to the complete descriptions within the owner's manual, refer to the Function Index in either volume of the owner's manual.

Note that usually you supply any needed operands *before* you execute the function (the operator). The exceptions are the *parameter functions*, which cue you for information *after* you execute the function. Parameter functions are shown below with their parameters, such as *nn*.

Function names printed in blue are *Alpha names* and use Alpha execution or User-keyboard execution. Function names printed in black or gold are *keyboard names*, and have keys for execution on the Normal keyboard.

Function	Definition
[←]	*Back arrow.* Deletion.
	Append to Alpha register.
[+] ([+])	*Plus.*
[-] ([-])	*Minus.*
[*] ([×])	*Multiplied by.*
[/] ([÷])	*Divided by.*
[1/X] ([1/x])	*Reciprocal.*
[10↑X] ( )	*Common exponential.*
[ABS]	*Absolute value.*
[ACOS] ( )	*Arc cosine.*
[ADATE]	*Alpha date.* Append date to Alpha reg.
[ADV]	*Advance* printer paper.
[ALENG]	*Alpha length.* No. of characters in Alpha reg.
[ALMCAT]	*Alarm catalog.*
[ALMNOW]	*Alarm now.* Activate oldest past-due conditional or control alarm.
[ALPHA]	Alpha keyboard toggle.
[ANUM]	*Alpha number.* Find first digit string in Alpha reg.
[AOFF]	*Alpha keyboard off.*
[AON]	*Alpha keyboard on.*

Function	Definition
APPCHR	*Append characters* to record in text file.
APPREC	*Append record* to text file.
ARCL *nn* (*nn*)	*Alpha recall.* Append reg. *nn* to Alpha reg.
ARCLREC	*Alpha recall record.* Append record to Alpha reg.
AROT	*Alpha rotate n* places.
ASHF	*Alpha shift* six characters to the left.
ASIN ( )	*Arc sine.*
ASN *name, key* ( )	*Assign* function or label to User key.
ASROOM	*ASCII room.* Bytes available in text file.
ASTO *nn* (*nn*)	*Alpha store.* Copy first six characters from Alpha reg. into reg. *nn*.
ATAN ( )	*Arc tangent.*
ATIME	*Alpha time.* Append time to Alpha reg.
ATIME24	*Alpha time 24-hour.* Append time to Alpha reg. in CLK24 format.
ATOX	*Alpha to X.* Shift leftmost character out of Alpha reg. and convert to its character code.
AVIEW ( )	*Alpha view.*
BEEP ( )	*Beeper.*
BST ( )	*Back step* through program lines.
CAT *n* (*n*)	List *catalog n* (1 to 6).
CF *nn* (*nn*)	*Clear flag nn* (00 to 29).
CHS (CHS)	*Change sign.*

Function	Definition
CLA ( )	*Clear Alpha.*
CLALMA	*Clear alarm by Alpha.* Clear alarm whose message matches Alpha reg.
CLALMX	*Clear alarm by X.* Clear *n*th alarm.
CLD	*Clear display* of message.
CLFL	*Clear file* named (text or data file).
CLK12	*Clock 12-hour* (format).
CLK24	*Clock 24-hour* (format).
CLKEYS	*Clear all User keys.*
CLKT	*Clock time only* (format).
CLKTD	*Clock time and date* (format).
CLOCK	Display clock.
CLP *label*	*Clear program* specified by global label.
CLRALMS	*Clear all alarms.*
CLRG	*Clear all data registers.*
CLRGX	*Clear registers by X* (*bbb.eeeii*). Clear every *ii*th reg. from R_{bbb} through R_{eee}.
CLΣ ( )	*Clear summations.* Clear statistics regs.
CLST	*Clear stack.*
CLX ( )	*Clear X*-register (the usual display).
COPY	Copy ROM program specified by global label.
CORRECT	Set time and adjust accuracy factor.
COS (COS)	*Cosine.*
CRFLAS	*Create file-ASCII.* Create text file of given name and length.

Function	Definition
CRFLD	*Create file-data* of given name and length.
D-R	*Degrees to radians* conversion.
DATE	Value for the date.
DATE+	Add number of days (in X-register) to date (in Y-register) to find new date.
DDAYS	*Delta days.* Find number of days between dates in X- and Y-registers.
DEC	*Decimal.* Octal to decimal conversion.
DEG	*Degrees* mode set.
DEL *nnn*	*Delete nnn* program lines, incl. current line.
DELCHR	*Delete n characters* from current text file, starting at pointer.
DELREC	*Delete current record.*
DMY	*Day-month-year* format.
DOW	*Day of week* of the given date (0=Sun.).
DSE *nn*	*Decrement and skip if less than or equal.* Given *iiiii.fffcc* in R_{nn}, decrement *iiiii* by *cc* and skip next line if *iiiii* is now $\leq$ *fff*.
ED	*Text editor.*
EEX	*Enter exponent.*
EMDIR	*Extended memory directory* (catalog 4).
EMDIRX	*Extended memory directory by X.* Find *n*th file's name and type.
EMROOM	*Extended memory room.* No. of regs. available.
END	End of program.

Function	Definition
ENG *n* (*n*)	*Engineering* display. Use $n+1$ digits and powers of 10^{3n}.
ENTER↑ (ENTER↑)	Separate sequential numbers.
E↑X ( )	*Natural exponential.*
E↑X-1	For arguments close to zero.
FACT	Factorial.
FC? *nn*	*Flag nn clear*? If not, skip next line.
FC?C *nn*	*Flag nn clear*? *Clear flag nn.*
FIX *n* (*n*)	*Fixed-point* display with *n* decimal places.
FLSIZE	*File size* (registers) of given file.
FRC	*Fractional part.*
FS? *nn* (*nn*)	*Flag nn set*? If not, skip next line.
FS?C *nn*	*Flag nn set*? *Clear flag nn.*
GETAS	*Get ASCII.* Copy mass-storage text file.
GETKEY	After 10 sec., return key code of key pressed (0 if none).
GETKEYX	*Get key by X.* After given no. of sec., return keycode (Y-register) and character code (X-register).
GETP	*Get program.* Replace last program with program file named.
GETR	*Get all registers* from given data file and copy to main memory.
GETREC	*Get record* from current text file and copy to Alpha reg., starting at pointer *rrr.ccc.*

Function	Definition
[GETRX]	*Get registers by X (bbb.eee).* Copy regs. in current data file (starting at pointer) to R_{bbb} through R_{eee} in main memory.
[GETSUB]	*Get subroutine* from named file and copy into main memory.
[GETX]	*Get X*-value from current data-file reg.
[GRAD]	Set Grads mode.
[GTO] *label* (*label*)	*Go to.* Program branch to given label.
[.] *nnn*	*Go to (dot).* Move current line to line *nnn* or *global label.*
[.][.]	*Go to (dot dot).* Move current line to end of program memory and pack memory.
[HMS]	*To hours-minutes-seconds.* Convert from decimal hours.
[HMS+]	*Hours-minutes-seconds plus.* Add degrees or times.
[HMS-]	*Hours-minutes-seconds minus.* Subtract degrees or times.
[HR]	*To decimal hours.* Convert from HMS.
[INSCHR]	*Insert characters* from Alpha reg. into text file starting at pointer.
[INSREC]	*Insert record.* Copy from Alpha reg. to new record at pointer.
[INT]	*Integer part.*
[ISG] *nn* (*nn*)	*Increment and skip if greater.* Given *iiiii.fffcc* in R_{nn}, increment *iiiii* by *cc* and skip next line if *iiiii* is now > *fff.*

Function	Definition
[LASTX] ( )	Recall number from LAST X reg.
[LBL] *label* (*label*)	*Label.*
[LN] ([LN])	*Natural log.*
[LN1+X]	For arguments close to 1.
[LOG] ([LOG])	*Common log.*
[MDY]	*Month-day-year* format.
[MEAN]	Means of accumulated *x*- and *y*-values.
[MOD]	*y* mod *x*.
[OCT]	*Octal.* Decimal to octal conversion.
[OFF]	Turn off computer.
[ON]	*Continuous on.* (Cancels automatic turn-off.)
[ON]	On/off toggle.
[P-R] ( )	*Polar to rectangular* conversion. Enter θ, then r. Returns x in X-reg., y in Y-reg.
[PACK]	Pack program memory.
[PASN]	*Programmable assign.* See [ASN].
[PCLPS]	*Programmable clear-programs.* Clear program named and all following programs.
[%] ( )	*x* percent of *y*.
[%CH]	*Percent change* from *y* to *x*.
[PI] ( )	Value of π to nine decimal places.
[POSA]	*Position in Alpha.* Find position of string (specified in X-register) in Alpha reg.
[POSFL]	*Position in file.* Pointer value of string (specified in Alpha reg.) in text file.

Function	Definition
PRGM	*Program mode* toggle.
PROMPT	Display the message in Alpha reg. and stop program (allowing input).
PSE	*Pause.* Interrupt program for a second.
PSIZE	*Programmable size.* See SIZE.
PURFL	*Purge file* named.
R↓	*Roll up* stack.
R-D	*Radians to degrees* conversion.
R-P ( )	*Rectangular to polar* conversion. Enter *y*, then *x*. Returns *r* in X-reg., θ in Y-reg.
R/S	*Run/stop* program.
RAD	*Radians* mode.
RCL *nn* (RCL *nn*)	*Recall* (copy) value from R_{nn}.
RCLAF	*Recall accuracy factor* for clock.
RCLALM	*Recall alarm* parameters for alarm *n*.
RCLFLAG	*Recall flag* status of flags 00-43.
RCLPT	*Recall pointer* value for current file.
RCLPTA	*Recall pointer by Alpha.* Recall pointer value for file named.
RCLSW	*Recall stopwatch* time.
RDN (R↓)	*Roll down* stack.
REGMOVE	*Register move.* Given *sss.dddnnn*, copy *nnn* registers from R_{sss} on, to R_{ddd} on.
REGSWAP	*Register swap.* Given *sss.dddnnn*, swap *nnn* registers from R_{sss} on, with R_{ddd} on.
RESZFL	*Resize file* (text or data) as specified.

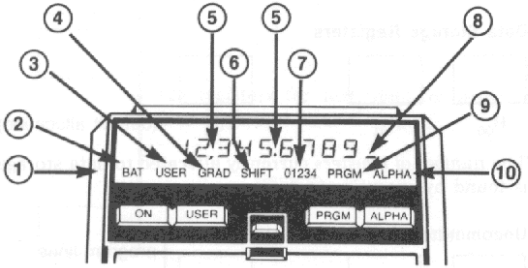
Function	Definition
RND	*Round.*
RTN ( )	*Return* program flow from subroutine to main program.
RUNSW	*Run stopwatch.*
SAVEAS	*Save ASCII.* Copy text file named to mass-storage file named.
SAVEP	*Save program* named to program file named.
SAVER	*Save all registers* in the given data file.
SAVERX	*Save registers by X (bbb.eee).* Copy R_{bbb} through R_{eee} to the current data file.
SAVEX	*Save x*-value in current data-file reg.
SCI *n* (*n*)	*Scientific* notation with *n* decimal places.
SDEV	*Standard deviations* of accumulated *x*- and *y*-values.
SEEKPT	*Seek pointer.* Set given pointer value for current text or data file.
SEEKPTA	*Seek pointer by Alpha.* Set given pointer value for the text or data file named.
SETAF	*Set accuracy factor* for clock.
SETDATE	*Set date* of clock.
SETIME	*Set time* of clock.
SETSW	*Set stopwatch* starting time.
SF *nn* (*nn*)	*Set flag nn* (00 to 29).
Σ+ (Σ+)	*Summation plus.* Add data value(s) to statistical accumulation.

Function	Definition
[Σ-] ( )	*Summation minus.* Delete data value(s) from statistical accumulation.
[ΣREG] *nn*	*Statistics registers* set to R*nn* through R*nn*+5.
[ΣREG?]	Find address of first statistics reg.
[SIGN]	1 or −1 for numbers, 0 for non-numbers, +1 for zero.
[SIN] ([SIN])	*Sine.*
[SIZE] *nnn*	Allocates *nnn* regs. to data storage.
[SIZE?]	No. of regs. allocated to data storage.
[SQRT] ([√x])	*Square root.*
[SST] ([SST])	*Single step* to next program line.
[ST+] *nn* ([STO] [+] *nn*)	*Store plus.* $R_{nn} + x$; result in R_{nn}.
[ST-] *nn* ([STO] [-] *nn*)	*Store minus.* $R_{nn} - x$; result in R_{nn}.
[ST*] *nn* ([STO] [×] *nn*)	*Store multiply.* $R_{nn} \times x$; result in R_{nn}.
[ST/] *nn* ([STO] [÷] *nn*)	*Store divide.* $R_{nn} \div x$; result in R_{nn}.
[STO] *nn* ([STO] *nn*)	*Store* copy of *x* in R_{nn}.
[STOFLAG]	*Restore flag* status of flags 00-43 from X-reg. Or: restore status of flags *bb* thru *ee* given *bb.ee* in X and flag data in Y.
[STOP] ([R/S])	Stop a running program.
[STOPSW]	*Stop stopwatch.*
[SW]	*Stopwatch.* Activate Stopwatch keyboard.

Function	Definition
[SWPT]	*Stopwatch and pointers.* Given *sss.rrr*, activate Stopwatch kbd. and set storage (*sss*) and recall (*rrr*) pointers.
[T+X]	*Time plus X.* Adjust time by increment given.
[TAN] ([TAN])	*Tangent.*
[TIME]	Value for the current time.
[TONE] *n*	$0 \leq n \leq 9$.
[USER]	User keyboard toggle.
[VIEW] *nn* (*nn*)	Display contents of R_{nn}.
[X↑2] ( )	*Square.*
[X = 0?] ( )	Conditional. If not true, skips next program line.
[X ≠ 0?]	
[X < 0?]	
[X <= 0?]	
[X > 0?]	
[X = Y?] ( )	
[X ≠ Y?]	
[X < Y?]	
[X <= Y?] ( )	
[X > Y?] ( )	
[X = NN?]	Conditional. Uses contents of R_{nn} (NN specified in Y-register) for comparison. If not true, skips next program line.
[X ≠ NN?]	
[X < NN?]	
[X <= NN?]	
[X > NN?]	
[X >= NN?]	
[X <>] *nn*	*X exchange* with R_{nn} contents.
[X <> F]	*X exchange flags* (status of flags 00-07).

Function	Definition
[X<>Y] ([x≷y])	*X exchange Y* contents.
[XEQ] *name* ([XEQ] *name*)	*Execute* given function or label.
[XTOA]	*X to Alpha.* Convert *x* (a character code) to equiv. character and append to Alpha reg.
[XYZALM]	*XYZ alarm* set (see page 32).
[Y↑X] ([y^x])	*y* to the *x* power (enter *y*, then *x*).

Display Features



1. **Display Annunciators.**
2. **Low-Power Condition.**
3. **User Keyboard Active.**
4. **Current Angular Mode.**
5. **Digit Separator and Radix Mark: Flag 28 set.**
   [CF] 28 reverses them.
   [CF] 29 removes the digit separator.
6. **Shift Set.**
   (To cancel, press ■ again.)
7. **Flag(s) Set**
   (flags 00 through 04).
8. **Input Cue.**
9. **Program Mode**
   or program running.
10. **Alpha Keyboard Active.**

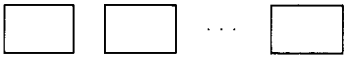
The display message **MEMORY LOST** indicates that Continuous Memory has been cleared and reset.

The program execution indicator, ▷ , appears and moves each time the program encounters a label.

Organization of Memory

Main Memory*

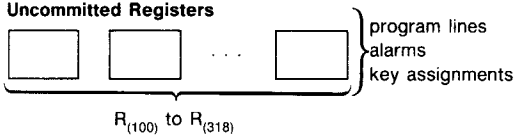
Data Storage Registers



(default allocation)

The *number of registers* currently allocated to data storage is found by executing [SIZE?].

Uncommitted Registers

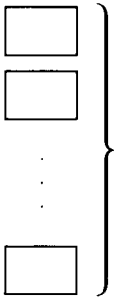


The number of uncommitted registers still available for use is displayed at the end of catalog 1 and after pressing [·][·] in Program mode.

Whenever Continuous Memory is cleared, R_{00} through R_{99} are allocated to data storage. This distribution of registers in main memory exists until you change it by executing [SIZE] *nnn* (where *nnn* is the number of registers to be in data storage).

* This diagram is simplified from the more complete one in section 12 of the owner's manual.

Extended Memory



124 registers for text, data, or program files of various sizes.

The number of registers still available in extended memory is displayed by [EMROOM] and at the end of catalog 4.

Storing and Executing Programs

To store a program in main memory:

1. Press [PRGM] to activate Program mode.
2. Press [·][·] to pack memory and move to the end of program memory.
3. Key in a global label of up to seven Alpha characters.
4. Key in each subsequent instruction.
5. Optional: press [·][·] to automatically add an [END] instruction and pack program memory.
6. Press [PRGM] to activate Execution mode.

If you make any mistakes, use [←] to delete individual characters and entire lines.

To execute a program in main memory:

1. Make sure Execution mode is active (no **PRGM** annunciator).
2. Start the program by executing its global label—by Alpha execution (page 14) or by User key (page 5 in this guide). Program execution *starts* at that global label.

While the program is actually running, the **PRGM** annunciator is on. The ⟩ program execution indicator also appears.

Pressing [R/S] will either start the current program (from its current line) or stop a running program. If a running program stops to prompt for data, for example, you key in the data and then press [R/S] to continue the program.

To run (and re-run) the current program, you can simply press [R/S].

Time and Alarm Formats

Time Values

The computer interprets clock time values that you specify according to the following conventions:

Time Settings

Setting	Clock Time
0	Midnight
1	1 (a.m.)
2	2
⋮	⋮
10	10
11	11
12	Noon
−1 or 13	1 p.m. or 13:00
−2 or 14	2 or 14
⋮	⋮
−10 or 22	10 or 22
−11 or 23	11 or 23
0	Midnight

Results of clock-time operations ([TIME], [RCLALM]) are always expressed in a 24-hour format in the X-register. Midnight is zero.

Alarm Format

Message Alarm: sounds tones and displays a message when it goes off.

Control Alarm: runs the specified program or programmable catalog-2 function when the alarm comes due.

Conditional Alarm: does not interrupt a running program, unlike the other alarms. If the HP-41CX is off or displaying the clock, a conditional alarm becomes a control alarm. If the HP-41CX is on and **not** running a program, a conditional alarm becomes a message alarm. If a program **is** running, the alarm only beeps (twice), and then becomes past due.

To set an alarm ([XYZALM]), follow these steps:

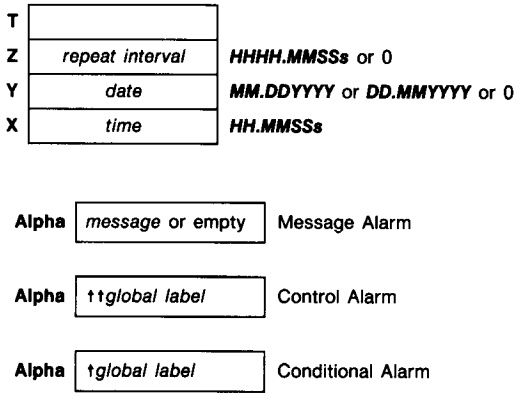
1. Key in the repeat interval (*using zero for no repetition*). Press [ENTER↑].
2. Key in the date for the alarm (*using zero for today*). Press [ENTER↑].
3. Key in the time for the alarm.
4. Press [ALPHA].

   For a message alarm, key in a message *or* clear the Alpha register. (A clear Alpha register results in an alarm "message" of the time and date.)

   For a control alarm, key in *global label* or *function name.*

   For a conditional alarm, key in *global label* or *function name.*

   Press [ALPHA] again.
5. Execute [XYZALM].



Acknowledging and Clearing Message Alarms

- To halt a current, flashing alarm, press any key *except* [STO]. This also clears (deletes) the alarm, unless it is a repeating one. A repeating message alarm is reset.
- To halt and clear a current repeating alarm, press [C].
- To clear an alarm that is *not* currently active, use [C] on the Alarm Catalog keyboard. (Run the catalog, stop it at the desired alarm, and press [C].)

You do not acknowledge non-message alarms, that is, ones that run programs.

The Catalogs

There are six catalogs (press *n*) in the HP-41CX:

- **Catalog 1: User Programs.** A list of all global labels and END instructions with the byte count for that program, listed in the order in which they were stored. The permanent END (**.END.**) shows the number of unused registers in uncommitted memory (and therefore still available for programming).
- **Catalog 2: External Functions + Time Functions + Extended Functions.** A list of all functions and programs currently available to the computer from peripheral devices, plug-in modules, and the time, extended, and extended-memory functions. The list of functions is grouped by source (press [ENTER↑] to see individual functions).
- **Catalog 3: Standard Functions.** An alphabetical list of the standard functions.
- **Catalog 4: Extended Memory Directory ([EMDIR]).** A list of all files in extended memory. It gives the file name, file type, and the number of registers in the file. It ends with the number of registers left in extended memory.
- **Catalog 5: Alarm Catalog ([ALMCAT]).** A list of each alarm, in chronological order, with its time, date, and message. (See the Alarm Catalog keyboard diagram.)
- **Catalog 6: User Key Assignments.** A list of all User key definitions in order of keycode.

When you execute *n*, the catalog listing begins. You can stop and restart it with [R/S]. With the automatic listing stopped, you can step through it forwards with [SST] and backwards with , or exit the catalog with [←]. In catalog 2, press [ENTER↑] to see a list of those functions belonging to the displayed source device.

Most automatic catalog listings speed up when you press an undefined key. If a printer is attached, the catalogs will print out in Trace mode only.

Character Codes

Code	ASCII	Display	Code	ASCII	Display
0			32	space	
1			33	!	
2			34	"	
3			35	#	
4			36	$	
5			37	%	
6			38	&	
7			39	'	
8			40	(	
9			41	)	
10			42	*	
11			43	+	
12			44	,	
13			45	-	
14			46	.	
15			47	/	
16			48	0	
17			49	1	
18			50	2	
19			51	3	
20			52	4	
21			53	5	
22			54	6	
23			55	7	
24			56	8	
25			57	9	
26			58	:	
27			59	;	
28			60	<	
29			61	=	
30			62	>	
31			63	?	

Code	ASCII	Display	Code	ASCII	Display
64	@		96		
65	A		97	a	
66	B		98	b	
67	C		99	c	
68	D		100	d	
69	E		101	e	
70	F		102	f	
71	G		103	g	
72	H		104	h	
73	I		105	i	
74	J		106	j	
75	K		107	k	
76	L		108	l	
77	M		109	m	
78	N		110	n	
79	O		111	o	
80	P		112	p	
81	Q		113	q	
82	R		114	r	
83	S		115	s	
84	T		116	t	
85	U		117	u	
86	V		118	v	
87	W		119	w	
88	X		120	x	
89	Y		121	y	
90	Z		122	z	
91	[		123	{	
92	\		124	|	
93	]		125	}	
94	^		126	~	
95	_		127		

The Flags and Their Status

0 = clear. ? = depends on other conditions.
1 = set. M = maintained by Continuous Memory.

Flag Number	Flag Name	Status at Reset, Turn-On
00–10	**User Flags**	0, M
	You can test and alter these flags.	
11–29	**Control Flags**	
	You can test and alter these flags.	
11	Automatic Execution	0, 0
12–20	External Device Control	0, 0
21	Printer Enable	?, ?
22	Numeric Data Input	0, 0
23	Alpha Data Input	0, 0
24	Range-Error Ignore	0, 0
25	Error Ignore	0, 0
26	Audio Enable	1, 1
27	User Keyboard	0, M
28	Radix Mark	1, M
29	Digit Separator Mark	1, M
30–55	**System Flags**	
	You can test but not alter these flags.	
31	Date Format	0, M
36	Number of Digits	0, M
37	"	1, M
38	"	0, M
39	"	0, M
40	Display Format	1, M
41	"	0, M
42	Grads Mode	0, M
43	Radians Mode	0, M
44	Continuous On	0, 0
48	Alpha Keyboard	0, 0
49	Low Power	?, ?
50	Message	0, 0
55	Printer Existence	?, ?

List of Errors

Following is a simplified description of each error message. For complete descriptions of the error conditions, refer to appendix A in the owner's manual. The function that caused an error does not get executed. You can clear an error message by pressing [←].

Error	Meaning
ALPHA DATA	Nonnumeric data used.
CHKSUM ERR	Part of file lost.
DATA ERROR	Illegal operand.
DUP FL	A file of that name already exists.
END OF FL	Pointer is at end of file.
END OF REC	Pointer is at end of record.
ERROR *Dnn*	Number not in time format.
ERROR *Rnn*	Number greater than 99.
FL NOT FOUND	Specified file does not exist.
FL SIZE ERR	Invalid file size.
FL TYPE ERR	Invalid file type.
KEYCODE ERR	Nonassignable keycode.
MEMORY LOST	Continuous Memory has been cleared and reset.
NAME ERR	Invalid file name.
NO DRIVE	The necessary device absent.
NONEXISTENT	The register, label, or function specified does not exist.
NO ROOM	Not enough room in memory.
NO SUCH ALM	Alarm does not exist.
OUT OF RANGE	Number too large.
PRIVATE	Program on card or cassette is private.

(table continued next page)

RAM	The global label specified already exists in main memory.
REC TOO LONG	Record too long.
ROM	You cannot modify a program in ROM.

HEWLETT PACKARD

Portable Computer Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

European Headquarters
150, Route Du Nant-D'Avril
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland

HP-United Kingdom
(Pinewood)
GB-Nine Mile Ride, Wokingham
Berkshire RG11 3LL

00041-90475 English
Printed in Singapore 8/83