Keyboard layout:

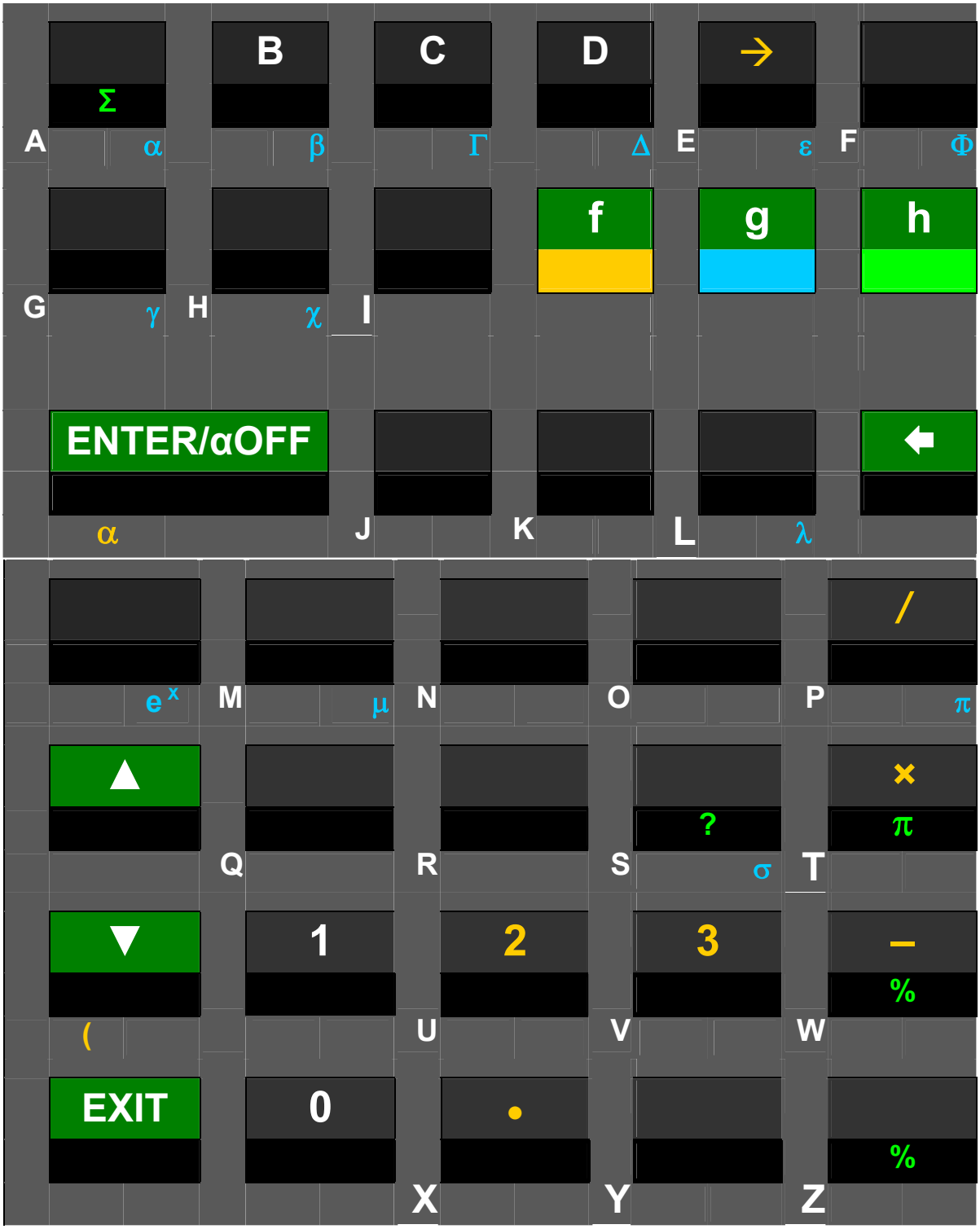
- CPX may be combined with many arithmetic and transcendental functions.
- → does conversions combined with H, H.MS, DEG, RAD, GRAD, 2, 8, 10, 16.
- The keys C, D, and E immediately call the respective user programs if defined.
- Calculator modes are as described in the paragraph about indicators below.
- Labels calling catalogues are underlined.

Please see the index for more.

Virtual active keyboard in hexadecimal mode. [→] is for addressing and temporary display in another base only (see the index below). Primary functions of the top 6 keys will be numeric input, so their default primary functions are accessed using [f] .
– In the other integer modes, the active keyboard will look alike, but the keys not needed for numeric input in the respective base will keep their default primary functions, except [Σ+] and [CPX] .

Virtual active keyboard in alpha mode. Therein, the 15 bytes alpha register is displayed in the dot matrix and all input is appended directly to this register – the numeric line is accessible by commands only. If a label is printed on green or red background in this picture it executes the respective function, else it calls an alpha catalogue or adds a character to the alpha register directly. Primary function of most keys will be adding the dark red letter printed bottom left of such a key. There, f becomes necessary then for reaching the function on key top, and g leads to homonymic Greek characters, where applicable. There are 3 exceptions, **ψ** being accessed via g 0 (below PSE), **τ** via g − (one key below **T**), and **η** via f EEX.

The catalogues f →, f CPX, f 1, h STAT, and h ./, offer even more characters (see the table below). Generally, ⬆ toggles upper and lower case, and PSE inserts a space.



Virtual active keyboard in temporary alpha mode. This mode is entered when opening catalogues and during some addressing. The functions printed on green background allow for catalogue browsing, error recovery, and leaving. See regular alpha mode, addressing tables, and catalogues for more.

MEMORY

Stack

L	**I**

T
Z
Y
X

Display

In complex arithmetic, stack registers **X** and **Y** contain the real and imaginary part of the first complex number, while **Z** and **T** carry the second. Stack register **I** takes the imaginary part of the last argument if a complex function is used. See LASTx.
As long as no complex function is used, **I** may be taken as another general purpose register.

General purpose registers 01 and 02 may be used for some statistical distributions. Registers 87 through 99 take the statistical sums indicated as soon as [Σ+] is used.

Flag D is set if “NaN” and “infinite” are allowed as results of commands.

General purpose registers

00
01
02
…
…
86
87 Σ ln x
88 Σ ln² x
89 Σ ln y
90 Σ ln² y
91 Σ (ln x · ln y)
92 Σ (x ln y)
93 Σ (y ln x)
94 n
95 Σ x
96 Σ x²
97 Σ y
98 Σ y²
99 Σ xy

Flags

00
01
02
…
…
97
98
99
B Overflow
C Carry
D Danger

Alpha

Program memory

001
002
003
…
…
498
499
500

ADDRESSING REGISTERS

1	User input	x< ?, x≤ ?, x= ?, x≠ ?, x≥ ?, and x> ?				RCL, STO, aRCL, aSTO, VIEW, x<>, DSE, ISG, DSZ, ISZ, FIX, SCI, ENG, DISP, BASE, CF and the other flag commands, CB and many more bit commands		
	Display	OP _ (e.g. x > _) Temporary alpha mode is on.				OP _ (e.g. RCL _) Alpha mode is off.		
2	User input [1]	0 or 1	X, Y, Z, T, L, or I	ENTER↑ [2] closes alpha.	→ closes alpha.	X, Y, Z, T, L, or I [3]	*Number of register or flag or bit(s) or decimals* [4]	→
	Display	OP *n* e.g. x ≤ 0 ?	OP *x* e.g. x ≥ y ?	OP r_	OP →_	OP s *x* e.g. SCI sZ	OP *nn* e.g. SF 15	OP →_ (indirect addressing)
3	User input	Compares *x* with the number **0**.	Compares *x* with the number on stack level **Y**.	*Register no.* 0 0 … 9 9	Look right for more about indirect addressing.	Sets scientific display with the number of decimals specified in stack level **Z**.	X, Y, Z, T, L, or I	*Register number* 0 0 … 9 9
	Display			OP r *nn* e.g. x ≠ r23?			OP →s *x* e.g. VIEW →sL	OP → *nn* e.g. STO →45
				Compares *x* with the number in reg. **23**.			Shows the content of the register where **LASTx** is pointing to.	Stores *x* into the register where register **45** is pointing to..

[1] For RCL and STO, any of +, −, ×, /, ▲, or ▼ may precede step 2. See the index of operations. RCL Σ+ recalls Σx and Σy.

[2] This step may be skipped for register numbers > 19. Input of any digit >1 will close alpha mode implicitely.

[3] For RCL, STO, VIEW, and x<> only. The stack registers may be addressed directly without explicitly switching to alpha mode before, except for RCL and STO, where a preceding ENTER↑ is necessary for the registers **Z** and **T**. Some stack operations may be useless but allowed, e.g. **x<>sX** . It is the user's responsibility not to mix pairs in complex mode, since a complex operation will always affect two registers: the one specified and the adjacent register.

[4] Generally, register and flag numbers may be 00 … 99, number of decimals 0 … 11, bases 2 … 16, bit numbers up to 63, integer word size up to 64. For numbers <10, you may key in e.g. 5 ENTER↑ instead of 0 5 . There are three additional flags addressed via B, C, and D .

ADDRESSING LABELS

1	User input	GTO, XEQ, LBL, Σ, π, SOLVE or INTEG [5]			
	Display	OP _ (e.g. GTO _) Alpha mode is off.			
2	User input	B, C, or D	ENTER↑ turns alpha mode on.	→ [6]	*2-digit numeric label* 0 0 … 9 9
	Display	OP '*name*' e.g. GTO'B'	OP '_	OP →_ (indirect addressing)	OP *nn* e.g. LBL 07
3	User input		*Label* [7] + ENTER↑ Last key closes alpha mode.	X, Y, Z, T, L, or I [8]	*Register number* 0 0 … 9 9
	Display		OP '*name*' e.g. SLV'STF'	OP →s *x* e.g. INT →sY	OP → *nn* e.g. XEQ →44
			Solve the function **STF** (with STF keyed in).	Integrate the function which's label is on stack level **y**.	Execute the routine which's label is in register **44**.

[5] SOLVE and INTEG will be displayed and listed as SLV and INT, respectively. The routines labelled B, C, and D may be called for execution directly via B, C, or D, respectively, without pressing XEQ before.

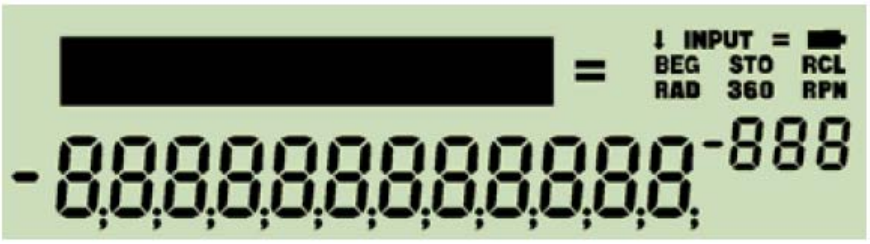
[6] Works with all these operations except LBL.

[7] Such a label may consist of up to 3 alphanumeric characters. ENTER↑ is only needed if less than 3 characters are entered.

[8] You may address these registers directly without switching to alpha mode before.

DISPLAY

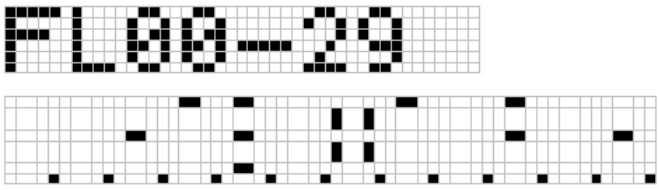
The display features 3 sections: numeric, dot matrix and fixed symbols. The latter are called annunciators, and are for indicating modes (see the paragraph about indicators below). The dot matrix is 6 dots high and 43 dots wide, allowing for some 7 to 12 characters, depending on their widths. The numeric section features a sign, 12 digits for the mantissa, a sign for the exponent, and 3 digits for the exponent.



In general, the 34S uses the dot matrix for indication of some more modes than the annunciators allow, for showing the alpha register, and for passing additional information to the user.

Some commands and modes use the display sections in a special way:

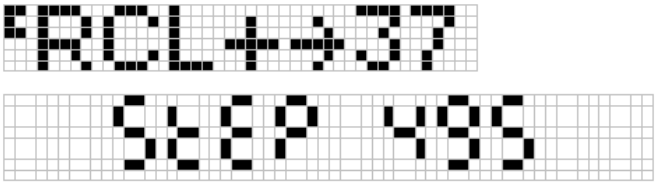
1. STATUS displays the status of the flags in a very compact way, allowing a brief status overlook after some training. For example, if the flags 2, 3, 5, 7, 11, 13, 17, 19, and 23 are set, calling STATUS will display this:



   Within the numeric section, each row of horizontal bars in the mantissa shows the status of 10 flags. When a flag is set, the respective bar turns black. So here the top row indicates flags 0 and 1 are clear, flags 2 and 3 are set, and flag 4 is clear. Then, a divider separates the first group of five flags from the next. Thereafter, flag 5 is set, 6 is clear, 7 is set, 8 and 9 are clear. Next row starts with flag 10 being clear, 11 set, 12 clear, 13 set, etc. Scrolling down by ▼ will display flags 10 – 39, then 20 – 49 etc. until 90 – D. Scrolling up by ▲ reverts this. Alternatively, pressing a digit will show the flags starting with 10 times this digit. The numeric exponent displays the status of the 3 hotkeys top left on the keyboard: if **b**, **c** or **d** show up, the respective label is defined in program memory.
   The STATUS display will disappear when another key is pressed but ▼, ▲ or a digit.

2. In integer modes, word size and complement setting are shown in the dot matrix, while the exponent indicates the base setting, carry, and overflow (see next paragraph).

3. In programming mode, the numeric display indicates the program step (001 – 500) in the mantissa and the number of unused steps in the exponent, while the dot matrix shows the command contained in the respective step.



INDICATORS

There are a number of indicators signaling the mode the calculator is running in.

Indicator	*INPUT*	b	d	h	o	*STO*
Mode name if different	α					**PRG**
Set by operation	αON [9]	BASE 2	BASE 10	BASE 16	BASE 8	PRGON
Cleared by operation	αOFF	BASE ≠2 FLOAT FRACT	BASE ≠10 FLOAT FRACT	BASE ≠16 FLOAT FRACT	BASE ≠8 FLOAT FRACT	PRGOFF

Indicator	*360*	G	H.MS	*RAD*	/c
Set by operation	DEG	GRAD	H.MS >H.MS	RAD	BASE 1 FRACT 2nd . in input (\HMS)
Cleared by operation	GRAD RAD	DEG RAD	BASE FLOAT >HR	DEG GRAD	BASE ≠1 FLOAT

INPUT, STO, 360, and *RAD* are annunciators (see previous paragraph). Outside integer modes, everything else is indicated in the text line. A capital **C** shows the last operation executed was a complex one, so you know you have to look at ***x*** and ***y*** then. The different date modes are signaled by **D.MY** or **M.DY**. Defaults Y.MD and FLOAT are not indicated. RPN may be shown permanently.

Within integer modes, word size and complement setting are shown in the dot matrix in a format WW.C, with C being **1** or **2** for 1's or 2's complement, **U** for unsigned, or **S** for sign-and-mantissa mode. In these modes, the exponent is used for further indications: its sign and its first digit show the base, a "**c**" in the second digit signals a carry bit set, an "**o**" in the third an overflow. Integer bases are indicated as follows:

Base	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Display	b	3	4	5	6	7	o	9	d	-1	-2	-3	-4	-5	h

All inputs are interpreted according to the mode set at input time. – A running program is signaled by a flashing *RCL* annunciator.

[9] Temporary alpha mode may entered during addressing – see the tables above for more information. Temporary alpha mode will also be entered opening the catalogues X.FCN, P.FCN, STAT, CONST, or CONV to allow for easy finding and selecting the item you are interested in. Keying in a character will jump to the first item starting with it, entering a second one will jump to the first item starting with this sequence, or anything following in the alphabet if this is not found. ENTER↑ will select the item you are on. After executing it, mode will return to the state it had before calling the catalogue, except the command chosen, while executing, did change the mode.

INDEX OF OPERATIONS

This lists all functions available on the 34S with the necessary keystrokes. Functions accessible via X.FCN will show up in this catalogue with their names unless specified differently explicitly. Sorting is as follows: 0 … 9, A … Z, α … ω, (,), +, -, *, /, ±, ",", ".", !, ?, ⇔, ←, ↑, ↓, →, <, ≤, =, ≠, ≥, >, #, °, %, √, ∞. Super- and subscripts are handled like normal characters prefixed by ↑ or ↓, respectively. Sorting is case insensitive.

The operations will work like on the HP-42S, special bit and integer functions like on the HP-16C, unless stated otherwise under remarks. Functions available on the 34S for the first time on an RPN calculator are highlighted under remarks, as are functions deviating from known ones carrying the same name. If no parameters are specified though required, they will be taken from the stack. Modes are abbreviated by their indicators. In this column an "&" represents logical AND, and a backslash stands for "all but", so e.g. ABS works in all modes but alpha. General operations with modes printed on red background are not programmable.

Name in listings	Keys to press	Works in modes	Remarks
c…	CPX …	FLOAT	Indicates complex operations [10]. CPX may be combined with any function which's name is printed in *italics* in this table. The three dots will be replaced in the listing by the name of the function attached. E.g. CPX f COS will be listed as c**COS**.
10^x	g 10^x	FLOAT	
12h	h X.FCN 12h	FLOAT	Sets 12h time display.
1COMPL	h X.FCN 1COMPL	Integer	Sets 1's complement mode like in HP-16C.
1/x	f $1/x$	FLOAT	
	B	FLOAT	Works as long as label B is not defined yet.
24h	h X.FCN 24h	FLOAT	Sets 24h time display.
2COMPL	h X.FCN 2COMPL	Integer	Sets 2's complement mode like in HP-16C.
2^x	g 2^x	\α	
ABS	f |x|	\α	
	CPX f |x|	FLOAT	Returns the magnitude $x = \sqrt{x^2 + y^2}$ and clears **Y**.

[10] Such operations work with pairs of adjacent registers. In each such pair, the first register contains the real and the second the imaginary part of the respective complex number. Unless stated otherwise explicitly, where a real function works with ***x*** (and maybe ***y***), its complex sibling works with ***x*** + *i* ***y*** (and maybe ***z*** + *i* ***t***). Where a real function works with a register at address ***a***, the respective complex function works with the registers at ***a*** and ***a*** + *1*.

Name in listings	Keys to press	Works in modes	Remarks
ACOS	g COS^{-1}	FLOAT, H.MS	
ACOSH	g HYP^{-1} COS	FLOAT	
ALL	h X.FCN ALL	FLOAT	Selects “all” display format.
AND	h AND	Integer	
		FLOAT	Works like AND in HP-28S, i.e. ***x*** and ***y*** are interpreted before executing this operation. Zero is taken as “false”, any other real number is “true”.
ANGLE	h X.FCN ANGLE	FLOAT	Calculates the angle between the positive x-axis and the straight line connecting the origin with the point (***x, y***).
ASIN	g SIN^{-1}	FLOAT, H.MS	
ASINH	g HYP^{-1} SIN	FLOAT	
ASR	h X.FCN ASR ***n***	Integer	Works like *n* (1 … 64) consecutive ASRs on HP-16C. In indirect addressing *n* may be zero. See the table above for more.
ATAN	g TAN^{-1}	FLOAT, H.MS	
ATANH	g HYP^{-1} TAN	FLOAT	
BASE	h X.FCN BASE ***n***	\α	Sets the base for integer calculations, with $2 \le n \le 16$. Popular bases are directly accessible on the keyboard. Furthermore, BASE 0 calls FLOAT, and BASE 1 calls FRACT. See additional opportunities for *n* in the table above.
BASE2	f 2		
BASE8	g 8		
BASE10	f 10		
BASE16	g 16		
BC?	h P.FCN BC? ***n***	PRG & integer	Tests the *n*-th bit in ***x*** and executes the next program line if this bit is clear, else skips the next program line. See the opportunities for *n* in the table above.
BestF	h STAT BestF	FLOAT	Selects the best curve fit model, maximizing the correlation like BEST in HP-42S.
BS?	h P.FCN BS? ***n***	PRG & integer	Works in analogy to “BC?”.

Name in listings	Keys to press	Works in modes	Remarks
B(k)	h STAT B(k)	FLOAT	= BINOMDIST(***x; r01; r02;*** 1) in MS Excel, with *r01* containing the sample size, and *r02* the probability to find an error in the population. The inverse returns *k* for a given probability *p* .
$B^{-1}(p)$	h STAT $B^{-1}(p)$		
CB	h X.FCN CB ***n***	Integer	Clears the *n*-th bit in ***x*** . See opportunities for *n* in the table above.
CEIL	h X.FCN CEIL	FLOAT	Returns the smallest integer ≥ ***x*** .
CF	h CF ***n***	\α	Clears one flag, working in analogy to CB.
CLALL	h X.FCN CLALL	\PRG	Clears all registers and all programs after confirmation.
CLFLAG	h P.FCN CLFLAG	\PRG	Clears all user flags.
CLPR	h CLPR	PRG	Clears current program after confirmation.
		\PRG, \α	Clears active program after confirmation.
CLREG	h X.FCN CLREG	\α	Clears all general purpose registers.
CLSTK	0 h FILL	\α	Clears stack registers **X, Y, Z**, and **T**.
CLx	h CLX	\α	Clears **X**.
	←	\α	Clears **X** if no input is pending.
	CPX h CLx	FLOAT	Clears both **X** and **Y**.
CLα	f CL α	All	Clears the alpha register like CLA in HP-42S.
CLΣ	g CLΣ	FLOAT	Clears registers 87 through 99.
COMB	f $C_{y,x}$	FLOAT	Returns the number of possible sets of ***y*** items taken ***x*** at a time. No item occurs more than once in a set, and different orders of the same ***x*** items are not counted separately. Formula: $C_{y,x} = \binom{y}{x} = \frac{y!}{x!\cdot(y-x)!}$
CONJ	h CONJ	FLOAT	Changes the sign of ***y*** .
CORR	g r	FLOAT	Returns a correlation coefficient for the current statistical data and curve fitting model
COS	f COS	FLOAT, H.MS	
COSH	f HYP COS	FLOAT	

Name in listings	Keys to press	Works in modes	Remarks
DATE	h X.FCN DATE	FLOAT	Recalls the date from the real time clock and displays it in the date format selected. See D.MY, M.DY, and Y.MD. The function DATE in HP-12C corresponds to DAYS+ here (see below).
DAY	h X.FCN DAY	FLOAT	Takes ***x*** as a date and returns the day of week in the dot matrix and a corresponding integer in the numeric display (Sunday = 7).
DAYS+	h X.FCN DAYS+	FLOAT	Adds a number of days in **X** on a date in **Y** and displays the resulting date including the day of week (Sunday = 7). DAYS+ works like DATE in HP-12C.
DBLR	h X.FCN DBLR	Integer	Double precision commands like in HP-16C, but here for up to 128 bits.
DBL*	h X.FCN DBL*		
DBL/	h X.FCN DBL/		
DEG	g DEG	FLOAT	Sets angular mode to degrees.
DECOMP	h X.FCN DECOMP	/c	Decomposes the fraction in **X**, i.e. puts its numerator in **Y** and its denominator in **X**.
DENANY	h X.FCN DENANY	FLOAT	Sets default fraction format like in HP-32SII, allowing maximum precision.
DENFAC	h X.FCN DENFAC	FLOAT	Sets “factors of the maximum denominator” fraction format like in HP-32SII.
DENFIX	h X.FCN DENFIX	FLOAT	Sets fixed denominator fraction format like in HP-32SII.
DENMAX	h X.FCN DENMAX	FLOAT	Works as /c does in HP-32SII, but maximum (and default) denominator is 9999.
DISP	h X.FCN DISP ***n***	FLOAT	Changes the number of decimals while keeping the display format. See opportunities for *n* in the table above.
DROP	h P.FCN DROP	\α	Drops ***x***, changing stack contents from [***x, y, z, t***] to [***y, z, t, t***].
DROPY	h P.FCN DROPY	\α	Changes stack contents from [***x, y, z, t***] to [***x, z, t, t***].
DSE	f DSE ***r***	PRG	Given `cccccc.fffii` in *r*, this function decrements `cccccc` by `ii` and skips the next program line if `cccccc` is now ≤`fff` for DSE, or = 0 for DSZ. See opportunities for *r* in the table above.
DSZ	h P.FCN DSZ ***r***		
D.MY	h X.FCN D.MY	FLOAT	Sets the format for date calculations.

Name in listings	Keys to press	Works in modes	Remarks
D→J	h X.FCN D→J	FLOAT	Assumes ***x*** is a date and converts it to a Julian day number.
D→R	h X.FCN D→R	FLOAT	Assumes **X** containing degrees and converts them to radians. Angular mode is kept.
E3OFF	h X.FCN E3OFF	FLOAT	Toggles the thousands separator, being either a comma or a point depending on the radix setting.
E3ON	h X.FCN E3ON		
ENG	f ENG ***n***	FLOAT	Selects engineering display format. See opportunities for *n* in the table above.
ENTER↑	ENTER↑	\α	
ERF	h STAT ERF	FLOAT	Calculates the error function erf(***x***).
ExpF	h STAT ExpF	FLOAT	Selects the exponential curve fit model.
Ex(x)	h STAT Ex(x)	FLOAT	= EXPONDIST(***x***; ***r01***; 1) in MS Excel, with *r01* containing λ. The inverse returns *x* for a given probability *p*.
Ex $^{-1}$(p)	h STAT Ex $^{-1}$(p)		
e^x	g e^x	FLOAT	
e^x -1	h X.FCN e^x-1		
FB	h X.FCN FB ***n***	Integer	Inverts ("flips") the *n*-th bit in ***x***. See opportunities for *n* in the table above.
FC?	h P.FCN FC? ***n***	PRG	Tests the flag specified and executes the next program line if this flag is clear, else skips the next program line. Clears, flips, or sets this flag after testing, if applicable. See opportunities for *n* in the table above.
FC?C	h P.FCN FC?C ***n***		
FC?F	h P.FCN FC?F ***n***		
FC?S	h P.FCN FC?S ***n***		
FF	h X.FCN FF ***n***	\α	Flips the flag specified. See opportunities for *n* in the table above.
FIB	h X.FCN FIB	\α	Calculates the Fibonacci number F_x.
FILL	h FILL	\α	Copies ***x*** in **Y**, **Z**, and **T**.
FIX	f FIX ***n***	FLOAT	Selects fixed point display format. See opportunities for *n* in the table above.
FLOAT	f .d	\α	Works like DECM in HP-42S. Additionally, converts possible H.MS data in **X** to decimal.
	g H	H.MS	
FLOOR	h X.FCN FLOOR	FLOAT	Returns the largest integer ≤ ***x***.

Name in listings	Keys to press	Works in modes	Remarks
FP	g FP	FLOAT	Returns the fractional part of ***x*** .
FRACT	g b/c	FLOAT	Sets fraction mode like in HP-32SII. Maximum denominator is 9999. Absolute values must be > 10E-5 and < 10E5.
FS?	h FS? ***n***	PRG	Tests the flag specified and executes the next program line if this flag is set, else skips the next program line. Clears, flips, or sets this flag after testing, if applicable. See opportunities for *n* in the table above.
FS?C	h P.FCN FS?C ***n***		
FS?F	h P.FCN FS?F ***n***		
FS?S	h P.FCN FS?S ***n***		
F(x)	h STAT F(x)	FLOAT	= FDIST(***x***; ***r01***; ***r02***) in MS Excel, like Q(F) in HP-21S.
$F^{-1}(p)$	h STAT $F^{-1}(p)$	FLOAT	= FINV(***x***; ***r01***; ***r02***) in MS Excel, like F_P in HP-21S.
GCD	h X.FCN GCD	\α	Returns the Greatest Common Divisor of ***x*** and ***y*** .
Ge(k)	h STAT Ge(k)	FLOAT	Geometric distribution, returns $1-(1-p)^k$. The probability *p* to find an error in the population must be stored in *r01* . The inverse returns *k* for a given probability *p* .
$Ge^{-1}(p)$	h STAT $Ge^{-1}(p)$		
GRAD	g GRAD	FLOAT	Sets angular mode to gon or grads.
GTO	h GTO ***label***	PRG	Like in HP-32S. See opportunities for *label* in the table above.
		\PRG, \α	
	h GTO . ***label***	\PRG, \α	
	h GTO . .		
H.MS	g H.MS	FLOAT	Sets H.MS mode for time calculations.
H.MS+	+	H.MS	
H.MS–	–		
INT	h INTEG ***label***	FLOAT	Integrate the function given in the routine specified. Lower and upper integration limit must be supplied in **Y** and **X**, respectively, when calling INT. See opportunities for *label* in the table above. Otherwise, the user interface is as in HP-15C.
IP	f IP	FLOAT	Returns the integer part of ***x*** .

Name in listings	Keys to press	Works in modes	Remarks
ISG	g ISG ***r***	PRG	Given `ccccc.fffii` in *r*, this function increments `ccccccc` by `ii` and skips the next program line if `ccccccc` is now `>fff` for ISG, or = 0 for ISZ. See opportunities for *r* in the table above.
ISZ	h P.FCN ISZ ***r***		
I β	h X.FCN I β	FLOAT	Calculates the regularized incomplete beta function β(***x***, ***a***, ***b***) with ***a*** taken from **Z** and ***b*** from **Y**.
I Γ	h X.FCN I Γ	FLOAT	Calculates the regularized incomplete gamma function $\gamma(x,y)/\Gamma(x)$.
J→D	h X.FCN J→D	FLOAT	Assumes ***x*** is a Julian day number and converts it to a date.
LASTx	g LASTx	\α	Recalls ***I*** into **X**.
	CPX g LASTx	FLOAT	Recalls ***I*** and ***i*** into **X** and **Y**, respectively.
LBL	f LBL ***label***	PRG	Identifies programs and routines for execution and branching. See opportunities for *label* in the table above.
LCM	h X.FCN LCM	\α	Returns the Least Common Multiple of ***x*** and ***y***.
LEAP?	h P.FCN LEAP?	PRG & FLOAT	Takes ***x*** as a date, extracts the year, tests for a leap year and executes the next program line if true, else skips this line.
LinF	h STAT LinF	FLOAT	Selects the linear curve fit model.
LJ	h X.FCN LJ	Integer	
LN	f LN	FLOAT	
LN1+X	h X.FCN LN1+X		
LN β	h X.FCN LN β	FLOAT	Calculates the natural logarithm of *β* or Γ, respectively. See these functions.
LN Γ	h X.FCN LN Γ		
LogF	h STAT LogF	FLOAT	Selects the logarithmic curve fit model.
LOGy	f LOGy	FLOAT	Calculates the logarithm of ***x*** for base ***y***.
	CPX f LOGy	FLOAT	Calculates the logarithm of the complex number ***x*** + *i* ***y*** for base ***z*** + *i* ***t***.
LOG$_{10}$	f LG	FLOAT	
LOG$_2$	f LOG2	\α	Calculates the logarithm for base 2.

Name in listings	Keys to press	Works in modes	Remarks
LR	h L.R.	FLOAT	Calculates the parameters of the fit curve through the data points accumulated, according to the model selected. Returns *A0* in **X** and *A1* in **Y**. In the linear fit model, *A0* is the intercept and *A1* the slope of the regression line.
MASKL	h X.FCN MASKL ***n***	Integer	Work like MASKL and MASKR on HP-16C, but with the parameter following the command instead of taken from **X**. See opportunities for *n* in the table above.
MASKR	h X.FCN MASKR ***n***		
MAX	h X.FCN MAX	\α	Returns the maximum of ***x*** and ***y***.
MIN	h X.FCN MIN	\α	Returns the minimum of ***x*** and ***y***.
MIRROR	h X.FCN MIRROR	Integer	Reflects the bit pattern in ***x*** (e.g. 000101 → 101000 for word size 6).
MOD	h MOD	\α	MOD of HP-42S equals RMD of HP-16C.
M.DY	h X.FCN M.DY	FLOAT	Sets the format for date calculations.
NAND	h X.FCN NAND	Integer	
		FLOAT	Works in analogy to AND.
NaN?	h P.FCN NaN?	PRG	Tests ***x*** for "not a number" and executes the next program line if true, else skips it.
nBITS	h X.FCN nBITS	Integer	Counts bits set like #B on HP-16C.
NOP	h P.FCN NOP	PRG	
NOR	h X.FCN NOR	Integer	
		FLOAT	Works in analogy to AND.
NOT	h NOT	Integer	
		FLOAT	Works in analogy to AND.
nΣ	h STAT nΣ	FLOAT	Recalls the number of accumulated data points. Necessary for basic statistics.
N(x)	h STAT N(x)	FLOAT	= NORMDIST(***x***; ***r01***; ***r02***; 1) in MS Excel, with *r01* containing the mean value and *r02* the standard deviation.
$N^{-1}(p)$	h STAT $N^{-1}(p)$	FLOAT	= NORMINV(***x***; ***r01***; ***r02***) in MS Excel.
OFF	h P.FCN OFF	PRG	
ON	h P.FCN ON		
OR	h OR	Integer	
		FLOAT	Works in analogy to AND.

Name in listings	Keys to press	Works in modes	Remarks
PAUSE	h PSE	PRG	Pauses program execution for about 1 s.
PERM	g Py,x	FLOAT	Returns the number of possible arrangements of ***y*** items taken ***x*** at a time. No item occurs more than once in an arrangement, and different orders of the same ***x*** items are counted separately. Formula: $P_{y,x} = \frac{y!}{(y-x)!}$
PowerF	h STAT PowerF	FLOAT	Selects the power curve fit model.
PROMPT	h X.PCN PROMPT	PRG	Displays the Alpha register and halts program execution
P(k)	h STAT P(k)	FLOAT	= POISSON(***x***; ***r01*r02***; 1) in MS Excel, with *r01* containing the probability to find an error in the population, and *r02* the sample size. Alternatively, *r01* may contain the Poisson parameter λ if *r02* contains 1. The inverse returns *k* for a given probability *p* .
P $^{-1}$(p)	h STAT P $^{-1}$(p)		
Q(x)	f Q	FLOAT	= NORMSDIST(***x***; 1) in MS Excel, like Q in HP-32E and Q(z) in HP-21S.
Q $^{-1}$(p)	g Q^{-1}	FLOAT	= NORMSINV(***x***) in MS Excel, like Q^{-1} in HP-32E and z_P in HP-21S.
RAD	g RAD	FLOAT	Sets angular mode to radians.
RAND#	h STAT RAND#	\α	Returns a random number between 0 and 1 like RAN in HP-42S.
RCL	RCL ***s***	\α	Recalls *s* into ***x*** . See RCL+ for more.
RCLM	h X.FCN RCLM	PRG	Recalls important mode settings encoded in an integer, i.e. display mode and number of decimals, trig mode, time and date format, statistics fit mode, decimal radix and thousands separator settings.
RCLWS	h X.FCN RCLWS	Integer	Recalls the word size set. See WSIZE.

Name in listings	Keys to press	Works in modes	Remarks
RCL+	RCL + ***s***	\α	Recalls the content of address ***s*** , executes the specified operation on it and stores the result in **X**. RCL▲ (▼) takes the maximum (minimum) of the value in ***s*** and **X**. See opportunities for ***s*** in the table above. Complex RCL affects ***x*** and ***y*** as well as two general purpose registers as explained at the top of this table.
RCL–	RCL − ***s***		
RCL×	RCL × ***s***		
RCL/	RCL / ***s***		
RCL▲	RCL ▲ ***s***		
RCL▼	RCL ▼ ***s***		
RDX,	h .,/	FLOAT	Toggles the radix mark.
RDX.			
RJ	h X.FCN RJ	Integer	Works in analogy to LJ.
RL	h X.FCN RL ***n***	Integer	Works like ***n*** consecutive RLs / RLCs on HP-16C. For RL, 1 ≤ ***n*** ≤ 63. For RLC, 1 ≤ ***n*** ≤ 64. In indirect addressing ***n*** may be zero. See the table above for more.
RLC	h X.FCN RLC ***n***		
RNDINT	h X.FCN RNDINT	FLOAT	Rounds ***x*** to next integer. ½ rounds to 1.
ROUND	g RND	FLOAT	Rounds ***x*** using the current display format, like RND in HP-42S.
		/c	Rounds ***x*** using the current denominator, like RND in HP-32SII.
RR	h X.FCN RR ***n***	Integer	Works like ***n*** consecutive RRs / RRCs on HP-16C. See RL / RLC for more.
RRC	h X.FCN RRC ***n***		
RTN	g RTN	PRG	In a running program, branches the program pointer back to the line following the most recent XEQ instruction. If there is no matching XEQ instruction, program execution halts. From the keyboard, RTN moves the program pointer to line 00 of the current program.
RTN+1	n/a	PRG	Internal support routine.
R-CLR	h X.FCN R-CLR	FLOAT	Interprets ***x*** in the form ss.nn. Clears the contents of ***nn*** registers starting with number ***ss*** . If **X** contains e.g. 34.56, R-CLR will clear registers 34 through 89.

Name in listings	Keys to press	Works in modes	Remarks
R-COPY	h X.FCN R-COPY	FLOAT	Interprets ***x*** in the form ss.nndd. Takes *nn* registers starting with number *ss* and copies their contents to *dd*. For example, if **X** contains 7.0345678, contents of registers 07 – 09 will be moved to registers 45 – 47, overwriting their old contents.
R-SORT	h X.FCN R-SORT	FLOAT	Interprets ***x*** in the form ss.nn. Sorts the contents of *nn* registers starting with number *ss*. Assume **X** contains 49.026 and registers 49 and 50 contain 1.2 and –3.4, respectively; then R-SORT will end with the contents of these 2 registers swapped.
R-SWAP	h X.FCN R-SWAP	FLOAT	Works like R-COPY but swaps the register contents of source and destination.
R/S	R/S	\PRG, \α	Runs a program (beginning at the current program line) or stops a running program.
R↑	h R↑	\α	Rotates the stack contents one step up or down, respectively.
R↓	R↓		
R→D	h X.FCN R→D	FLOAT	Assumes **X** containing radians and converts them to degrees. Angular mode is kept.
s	g s	FLOAT	Calculates s_x and s_y using the current statistical data.
SB	h X.FCN SB ***n***	Integer	Sets the *n*-th bit in ***x***. See opportunities for *n* in the table above.
SCI	f SCI ***n***	FLOAT	Selects scientific display format. See opportunities for *n* in the table above.
SEED	h STAT SEED	FLOAT	Stores a seed for the random number generator.
SERR	h STAT SERR	FLOAT	Calculates $s/\sqrt{n}$ and returns the respective values in **X** and **Y**.
SETDAT	h X.FCN SETDAT	FLOAT, H.MS	Sets the date or time, respectively, for the real time clock.
SETTIM	h X.FCN SETTIM		
SF	h SF ***n***	\α	Sets the flag specified. See opportunities for *n* in the table above.
SIGN	h X.FCN SIGN	\α	Returns 1 for ***x*** > 0, –1 for ***x*** < 0, and 0 for non-numbers.
	CPX h X.FCN SIGN	FLOAT	Returns the unit vector of ***x*** + *i* ***y*** in **X** and **Y**.

Name in listings	Keys to press	Works in modes	Remarks
SIGNMT	h X.FCN SIGNMT	Integer	Sets sign-and-mantissa mode for integers.
SIN	f SIN	FLOAT, H.MS	
SINC	h X.FCN SINC	FLOAT	Calculates $\frac{\sin(x)}{x}$.
SINH	f HYP SIN	FLOAT	
SL	h X.FCN SL ***n***	Integer	Works like *n* (1 … 64) consecutive SLs on HP-16C. In indirect addressing *n* may be zero. See the table above for more.
SLV	h SOLVE ***label***	FLOAT	Solves the equation f(x) = 0, with f(x) calculated by the routine specified. 2 initial estimates of the root must be supplied in **X** and **Y** when calling SLV. See opportunities for *label* in the table above. Otherwise, the user interface is as in HP-15C.
SR	h X.FCN SR ***n***	Integer	Works like *n* consecutive SRs on HP-16C. See SL for more.
STO	STO ***d***	\α	Stores ***x*** into *d* . See STO+ for more.
STOM	h X.FCN STOM	PRG	Reverts RCLM.
STOP	R/S	PRG	Stops program execution.
STO+	STO + ***r***	\α	Calls the content of address *r* , executes **OP *x*** on it and stores the result into said address. STO▲ (▼) takes the maximum (minimum) of the values in *r* and **X** and stores the result. See opportunities for *r* in the table above. Complex STO affects ***x*** and ***y*** as well as two general purpose registers as explained at the top of this table.
STO–	STO − ***r***		
STO×	STO × ***r***		
STO/	STO / ***r***		
STO▲	STO ▲ ***r***		
STO▼	STO ▼ ***r***		
SUM	RCL Σ+	FLOAT	Recalls ***Σx*** into **X** and ***Σy*** into **Y**.
TAN	f TAN	FLOAT, H.MS	
TANH	f HYP TAN	FLOAT	
TIME	h X.FCN TIME	FLOAT, H.MS	Recalls the time from the real time clock.
t(x)	h STAT t(x)	FLOAT	= TDIST(***x***; ***r01***; 1) in MS Excel, like Q(t) in HP-21S.

Name in listings	Keys to press	Works in modes	Remarks
$t^{-1}(p)$	h STAT $t^{-1}(p)$	FLOAT	= TINV(2**x*; ***r01***) in MS Excel, like tp in HP-21S.
UNSIGN	h X.FCN UNSIGN	Integer	Sets unsigned mode for integers.
VERS	h X.FCN VERS	All	Displays the firmware version.
VIEW	h VIEW ***r***	All	Views the contents of *r* . See opportunities for *r* in the table above.
W	h X.FCN W	FLOAT	Calculates Lambert's W for given ***x*** ≥ -1/e
W^{-1}	h X.FCN W^{-1}	FLOAT	Inverts W, i.e. calculates *x* for given W (≥ -1).
Wb(t)	h STAT Wb(t)	FLOAT	= WEIBULL(***x***; ***r01***; ***r02***; 1) in MS Excel, with *r01* containing the *shape parameter*, and *r02* the *characteristic lifetime*. The inverse returns *t* for a given probability *p* .
$Wb^{-1}(p)$	h STAT $Wb^{-1}(p)$		
WSIZE	h X.FCN WSIZE ***n***	Integer	Works like WSIZE on HP-16C, but with the parameter following the command instead of taken from **X**. WSIZE 0 will set the word size to maximum. See opportunities for *n* in the table above.
XEQ	XEQ ***label***	PRG	Calls the respective subroutine.
		\PRG, \α	Executes the respective program. See opportunities for *label* in the table above.
	B , C , or D (Prefix f may be necessary for accessing these hotkeys for integer bases > 10.)	PRG	Calls the respective subroutine, so e.g. XEQ C will be inserted in the program when C is pressed.
		\PRG, \α	Executes the respective program if defined.
XNOR	h X.FCN XNOR	Integer	
		FLOAT	Works in analogy to AND.
XOR	h XOR	Integer	
		FLOAT	Works in analogy to AND.
x !	h !	FLOAT	
***x*↔**	h x≷ ***r***	\α	Swaps the contents of **X** and *r* . See opportunities for *r* in the table above. Complex x↔ swaps the contents of **X** and **Y** with those of two general purpose registers as explained at the top of this table.

Name in listings	Keys to press	Works in modes	Remarks
x*↔*y	x≷y	\α	Swaps ***x*** and ***y***, performing Re↔Im if a complex operation was executed before.
	CPX x≷y	FLOAT	Swaps ***x*** with ***z*** and ***y*** with ***t***.
***x*²**	g x²	\α	
x → α	h X.FCN X→α	All	Appends a character (specified by the code in **X**) to the alpha register. If **X** contains an alpha string, appends the entire string, like XTOA in HP-42S.
x < … ?	f x<? ***a***	\α	Compares ***x*** with ***a***. See opportunities for ***a*** in the table above. The three dots will be replaced in the listing by ***a*** according to the examples given in said table. CPX f x=? ***a*** and CPX g x≠? ***a*** compare ***x*** and ***y*** with ***a***. Here, if ***a*** = … • 0 then ***x*+*iy*** will be compared with 0; • 1 then ***x*+*iy*** will be compared with 1; • Z then ***x*+*iy*** will be compared with ***z*+*it***; • L then ***x*+*iy*** will be compared with ***l*+*ii***; • ENTER↑ ***nn*** then ***x*** will be compared with the contents of register ***nn*** and ***y*** with those of register ***nn+1***.
x ≤ … ?	g x≤? ***a***		
x = … ?	f x=? ***a***		
x ≠ … ?	g x≠? ***a***		
x ≥ … ?	f x≥? ***a***		
x > … ?	g x>? ***a***		
$\underline{\bar{x}}$, $\underline{\bar{y}}$	f $\bar{x}$	FLOAT	Recalls $\frac{1}{n}\sum x$ into **X** and $\frac{1}{n}\sum y$ into **Y**.
$\underline{\bar{x}}$w	h STAT $\underline{\bar{x}}$w	FLOAT	Returns the weighted mean $\sum xy / \sum y$.
$\underline{\hat{x}}$	h STAT $\underline{\hat{x}}$	FLOAT	Predicts a forecast ***x*** for a given ***y*** according to the curve fit model chosen. See L.R. for more.
Y.MD	h X.FCN Y.MD	FLOAT	Sets the format for date calculations.
y*^*x	g y^x	\α	In integer modes ***x*** must be ≥ 0.
	C or f C	\α	Works as long as label C is not defined yet. f is needed for integer bases >12 only.
ŷ	f ŷ	FLOAT	Predicts a forecast ***y*** for a given ***x*** according to the curve fit model chosen. See L.R. for more.

Name in listings	Keys to press	Works in modes	Remarks
αAPP	h X.FCN αAPP ***char***	\α	Enters alpha mode for the input of 1 character, appends this to the alpha register, and returns to the mode set before. Eventually, this equals the sequence αON *char* αOFF.
αDATE	h X.FCN αDATE	FLOAT, α	Takes ***x*** as a date and appends it to the alpha register in the format selected.
αDAY	h X.FCN αDAY	FLOAT, α	Takes ***x*** as a date, recalls the name of the respective day and appends it to the alpha register.
αIP	h X.FCN αIP	All	Appends integer part of ***x*** to the alpha register, like AIP in HP-42S.
αLENG	h X.FCN αLENG	All	Returns the number of characters in the alpha register, like ALENG in HP-42S.
αMONTH	h X.FCN αMONTH	FLOAT, α	Takes ***x*** as a date, recalls the name of the month and appends it to the alpha register.
αRCL	f RCL ***r***	α	Interprets the contents of *r* as characters and appends them to the alpha register. See opportunities for *r* in the table above.
	h X.FCN αRCL ***r***	\α	
αRC#	h X.FCN αRC# ***r***	All	As αRCL, but αRC# interprets the contents of *r* as a number and appends this in current format to the alpha register.
αRL	h X.FCN αRL ***n***	All	Rotates the alpha register by *n* characters like AROT in HP-42S, but with a positive parameter trailing the command instead of taken from **X**. In indirect addressing *n* may be zero. See the table above for more.
αRR	h X.FCN αRR ***n***	All	Works like αRL but rotates to the right.
αSL	h X.FCN αSL ***n***	All	Shifts the *n* left-most characters out of the alpha register, similar to ASHF in HP-42S. In indirect addressing *n* may be zero. See the table above for more.
αSR	h X.FCN αSR ***n***	All	Works like αSR but shifts the *n* right-most characters.
αSTO	f STO ***r***	α	Stores the first 6 characters in the alpha register into *r*. See opportunities for *r* in the table above.
	h X.FCN αSTO ***r***	\α	
αTIME	h X.FCN αTIME	FLOAT, α	Takes ***x*** as a time HH.MMSS, converts it to a string in the format selected, and appends it to the alpha register.
αVIEW	h X.FCN αVIEW	\α	Displays the alpha register.

Name in listings	Keys to press	Works in modes	Remarks
α → x	h X.FCN α→X	All	Converts the left-most character in the alpha register to its character code (returned to **X**) and deletes the character, like ATOX in HP-42S.
β	h X.FCN β	FLOAT	Calculates Euler's Beta function B(***x***, ***y***).
Γ	h STAT Γ	FLOAT	
	h X.FCN Γ		
ΔDAYS	h X.FCN ΔDAYS	FLOAT	Calculates the number of days between 2 dates ***x*** and ***y*** . Works like in HP-12C.
Δ%	f Δ%	FLOAT	Calculates $100 \cdot \frac{x-y}{y}$ like %CH in HP-42S.
π	h π	FLOAT	
	CPX h π	FLOAT	Returns π in **X** and clears **Y** for using π in complex calculations.
Π	h P.FCN Π ***label***	PRG	Generates a product or a sum, respectively, based on the routine starting with the label specified. The loop control number is taken from **X** in the standard format `cccccc.fffii` like e.g. for DSE. See opportunities for *label* in the table above.
Σ	h P.FCN Σ ***label***		
σ	h STAT σ	FLOAT	Calculates $s \cdot \sqrt{\frac{n}{n-1}}$ and returns the respective values in **X** and **Y**.
$\Sigma \ln^2 x$	h STAT $\Sigma \ln^2 x$	FLOAT	Recalls the respective statistical sum. These sums are necessary for the curve fitting models beyond pure linear. Calling them by name enhances readability of programs significantly.
$\Sigma \ln^2 y$	h STAT $\Sigma \ln^2 y$		
Σlnx	h STAT Σlnx		
Σlnxy	h STAT Σlnxy		
Σlny	h STAT Σlny		
Σxlny	h STAT Σxlny		
Σylnx	h STAT Σylnx		

Name in listings	Keys to press	Works in modes	Remarks
Σx	h STAT Σx	FLOAT	Recalls the respective statistical sum. These sums are necessary for basic statistics and linear curve fitting. Calling them by name enhances readability of programs significantly.
Σx^2	h STAT Σx^2		
Σxy	h STAT Σxy		
Σy	h STAT Σy		
Σy^2	h STAT Σy^2		
Σ+	Σ+	FLOAT	
Σ–	h Σ–		
χ^2(x)	h STAT χ^2(x)	FLOAT	= CHIDIST(***x***; ***r01***) in MS Excel, like Q(χ^2) in HP-21S.
χ^2INV	h STAT χ^2INV	FLOAT	= CHIINV(***x***; ***r01***) in MS Excel, like χ^2_p in HP-21S.
+, –, ×, /	+ , − , × , /	\α	
+/–	+/−		
//	g //	FLOAT	Calculates $\left(1/x + 1/y\right)^{-1}$.
%	g %	FLOAT	Calculates $x \cdot y/100$.
%T	h X.FCN h % T	FLOAT	Calculates $100 \cdot x/y$.
%Σ	h STAT h % Σ	FLOAT	Calculates $100 \cdot x/\sum x$.
%+	h %+	FLOAT	Calculates $x = y \cdot \left(1 + \frac{x}{100}\right)$, adding a markup of ***x*** % to ***y***.
%–	h %–	FLOAT	Calculates $x = y \cdot \left(1 - \frac{x}{100}\right)$, subtracting a discount of ***x*** % from ***y***.
∞?	h P.FCN ∞?	PRG	Tests ***x*** for infinity and executes the next program line if true, else skips this line.
√‾	f √x	\α	
	D or f D	\α	Works as long as label D is not defined yet. f is needed for integer bases >13 only.

Name in listings	Keys to press	Works in modes	Remarks
→DEG	→ g DEG	FLOAT	Assumes **X** containing an angle in current angular mode and converts it to degrees. Angular mode is kept.
→GRAD	→ g GRAD	FLOAT	Works like →DEG, but converts to gon or grads.
→HR	→ f H	H.MS	Takes the hours or degrees in **X** and converts them into decimal numbers.
→H.MS	→ g H.MS	FLOAT	Assumes **X** containing *decimal* hours or degrees and converts them into the format HHH.MMSS.
→POL	g ▶P	FLOAT	Assumes **X** and **Y** containing the coordinates (***x, y***) and converts them to (r , ϑ) .
→RAD	→ g RAD	FLOAT, H.MS	Works like →DEG, but converts to radians.
→REC	f ▶R	FLOAT	Assumes **X** and **Y** containing the coordinates (r , ϑ) and converts them to (***x, y***) .
			Pure input commands:
0 … 9	0 … 9	\α	Numeric input. The 6 top keys on the keyboard will be used for input of hexadecimal numbers >10 in HEXM by default. For bases <16, their defaults are switched as applicable. For bases <10, input of illegal digits will result in an error message.
A … F	A …F (dark red print on keyboard)	-1, -2, -3, -4, -5, h	
A … Z	A …Z (dark red print on keyboard)	α	Alphabetic input. See page 3 for more information. Find alpha menus below.
	f ⇧	α	Toggles upper and lower case.
	←	Input pending	Deletes last digit or character put in.
DEL	←	PRG	Deletes current step if no input is pending.
E	E (the key)	FLOAT	Like EEX in vintage calculators.
[ ] or **[/]**	,	/c	First , is taken as a space, second as a fraction mark, e.g. 2 , 3 , 4 results in 2 ¾ in the dot matrix display.
[.] or **[,]**	,	FLOAT	Inserts the radix mark as selected.
		α	Inserts a point.
[.]	,	Y.MD, D.MY, M.DY	Separates the leading unit in date modes. It is left to the user to decide if a number displayed represents a date or not.
[°]	,	H.MS	Separates degrees (or hours) from minutes and seconds, so input format is HH,MMSS.

Name in listings	Keys to press	Works in modes	Remarks
			Catalogues (non programmable): calling any catalogue will set alpha mode to allow for keying in the first 2 characters of the item wanted. In general, ▲ and ▼ browse the catalogue, and EXIT will leave it without executing anything. See next rows for the contents of the individual catalogues.
ARROWS	f →	α	Calls the catalogue of arrows and mathematical symbols. ENTER↑ inserts the character selected. This applies also to the other alpha catalogues except X.FCN.
COMPARE	f x <	α	For comparison symbols and brackets.
CONST	h CONS	FLOAT	Calls the catalogue of constants like in HP35s. See the list of constants stored in a separate table below. ENTER↑ recalls the constant selected into **X**.
	CPX h CONS	FLOAT	Works as above, but clears **Y** in recalling.
CONV	h CONV	FLOAT	Calls the catalogue of conversions. The conversions stored are listed in a separate table below. ENTER↑ converts ***x*** according to the conversion selected.
CPX	f CPX	α	For complex letters, necessary for languages beyond English. Upper or lower case will be displayed according to setting (see f ⇧ below).
PUNCT	h ./,	α	For punctuation marks and text symbols.
P.FCN	h P.FCN	\α	Calls the catalogue of extra programming functions. ENTER↑ executes the function selected (applies to STAT and X.FCN, too).
STAT	h STAT	FLOAT	For extra statistical functions.
		α	Contains some letters for statistics.
X.FCN	h X.FCN	FLOAT	For extra real functions.
		Integer	For extra integer functions.
		α	For extra alpha functions.
	CPX h X.FCN	FLOAT	For extra complex functions.

Name in listings	Keys to press	Works in modes	Remarks
			Pure navigation, mode switching and information commands (all non programmable):
	g OFF	All	
	ON	Calc. off	
	▲ / ▼	Status open	Go to previous / next set of flags.
		Catalogue open	Go to previous / next item in this catalogue.
		α	Move the cursor 1 character to the left / right in alpha register. Shift the display window if necessary.
		PRG	Like BST / SST in HP-42S.
		\PRG, \α	
	f ◀ / g ▶	Integer	Shift the display window like in HP-16C. Useful for numbers with small bases.
EXIT	EXIT	All	Exits catalogues and other operations with pending input, canceling the execution of this operation.
PRGOFF	h P/R	PRG	Toggle programming mode.
PRGON	h P/R	\PRG, \α	
SHOW	h SHOW	FLOAT, \PRG	Shows the full mantissa.
		PRG	Displays a CRC-32 checksum of program memory's contents (8 hex digits), allowing to validate program integrity.
STATUS	h STATUS	All	Shows the flag status, similar to STATUS on HP-16C. See the paragraph about display above for more.
αOFF	ENTER↑	α	Toggle alpha mode like AOFF and AON in HP-42S.
αON	f α	\α	
→BIN	→ f 2	\α, \h, \-5	Show ***x*** in target integer representation until the next command is executed. Mode is kept. In modes –5 and h, an f must precede the key →.
→DEC	→ f 10		
→HEX	→ g 16		
→OCT	→ g 8		

CATALOGUE CONTENTS

Here the contents of the catalogues X.FCN, P.FCN and STAT are listed. A single operation, e.g. BASE, may be contained in more than one catalogue. The characters necessary to get to a specific function in the catalogue are printed bold in this table – each red character must be replaced by a ▼ – if even the last letter of a function name is red, one may need more strokes of ▼ to access this function. The alpha catalogues ARROWS, COMPARE, CPX, and PUNCT are found further below. See the catalogues CONST and CONV in separate paragraphs.

Content of X.FCN …	… in FLOAT	CPX X.FCN	… in integer modes	… in alpha mode	Content of P.FCN	Content of STAT
12h	X		**1**COMPL	**C**LALL	**B**C?	**B**estF
24h	X		**2**COMPL	**V**ERS	**BS**?	**B(**k)
ALL	X		**A**SR	**x**→ α	**C**B	**B** $^{-1}$(p)
ANGLE	X		**B**ASE	αDATE	**CL**FLAG	**E**RF
BASE	X		**C**B	α**D**AY	**D**ROP	**Ex**pF
CEIL	X		**CL**ALL	α**I**P	**DR**OPY	**Ex(**x)
CLALL	X		**CL**REG	α**L**ENG	**DS**Z	**Ex** $^{-1}$**(**p)
CLREG	X		**D**BLR	α**M**ONTH	**F**B	**F**(x)
DATE	X		**DB**L*	α**R**C#	**FC**?	**F** $^{-1}$(p)
DAY	X		**DBL**/	α**RL**	**FC?**C	**G**e(k)
DAYS+	X		**F**B	α**RR**	**FC?F**	**Ge** $^{-1}$(p)
DECOMP	X		**FF**	α**S**L	**FC?S**	**L**inF
DENANY	X		**FI**B	α**SR**	**FF**	**Lo**gF
DENFAC	X		**G**CD	α→ x	**FS**?C	**n**Σ
DENFIX	X		**L**CM		**FS?**F	**N(**x)
DENMAX	X		**M**ASKL		**FS?S**	**N** $^{-1}$(p)
DISP	X		**MA**SKR		**I**SZ	**P**owerF
D.MY	X		**MAX**		**L**EAP?	**P(**k)
D→J	X		**MI**N		**N**aN?	**P** $^{-1}$(p)
D→R	X		**MI**RROR		**NO**P	**R**AND#
E3OFF	X		**N**AND		**O**FF	**S**EED
E3ON	X		**NB**ITS		**ON**	**SE**RR
e^{x} -1	X	X	**NO**R		**P**ROMPT	**t**(x)
FF	X		**R**AND#		**R**CLM	**t** $^{-1}$(p)

Content of X.FCN …	… in FLOAT	CPX X.FCN
FIB	X	X
FLOOR	X	
GCD	X	
I β	X	
I Γ	X	
J→D	X	
LCM	X	
LN1+x	X	X
LN β	X	X
LN Γ	X	X
MAX	X	
MIN	X	
M.DY	X	
NAND	X	
NOR	X	
RNDINT	X	
R→D	X	
SETDAT	X	
SETTIM	X	
SIGN	X	X
SINC	X	X
TIME	X	
VERS	X	
W	X	X
W $^{-1}$	X	X
XNOR	X	
x→ α	X	
Y.MD	X	
αAPP	X	

… in inte-ger modes
RCLWS
RJ
RL
RLC
RR
RRC
SB
SIGN
SIGNMT
SL
SR
UNSIGN
VERS
WSIZE
XNOR
x→ α
αAPP
α**I**P
α**L**ENG
α**O**N
α**R**CL
α**RC**#
α**RL**
α**RR**
α**S**L
α**SR**
α**STO**
α**V**IEW
α→ x

… in alpha mode

Content of P.FCN
R-CLR
R-COPY
R-SORT
R-SWAP
SB
STOM
Π
Σ
∞?

Content of STAT
Wb(t)
Wb $^{-1}$(p)
$\bar{\underline{x}}$w
$\bar{\underline{x}}$
Γ
σ
Σln^{2}x
Σln^{2}y
Σlnx
Σlnxy
Σlny
Σx
Σx^2
Σxlny
Σxy
Σy
Σy^2
Σylnx
χ^2(x)
χ^2INV
%Σ

Content of X.FCN …	… in FLOAT	CPX X.FCN	… in integer modes	… in alpha mode	Content of P.FCN	Content of STAT
αDATE	X					
αDAY	X					
αIP	X					
αLENG	X					
αMONTH	X					
αON	X					
αRCL	X					
αRC#	X					
αRL	X					
αRR	X					
αSL	X					
αSR	X					
αSTO	X					
αTIME	X					
αVIEW	X					
α→ x	X					
β	X	X				
Γ	X	X				
ΔDAYS	X					
%T	X					

Here are the contents of the alpha catalogues:

STAT
$\hat{x}$
$\bar{x}$
$\hat{y}$
$\bar{y}$

ARROWS
→
←
↑
↓
↔
√
∫
°
$\square^{-1}$
$\square^{2}$
$\hbar$
$\square^{x}$
^
∞

CPX	
À	à
Á	á
Â Ã Ā Ă	â ã ā ă
Ä	ä
Å	å
Ć	ć
Č	č
Ç	ç
È	è
É	é
Ê Ē Ĕ Ě	ê ē ĕ ě
Ë	ë
Ì	ì
Í	í
Î Ĩ Ī Ĭ	î ĩ ī ĭ
Ï	ï
Ñ	ñ
Ò	ò
Ó	ó
Ô Õ Ō Ŏ	ô õ ō ŏ
Ö	ö
Ř	ř
Š	š
	ß
Ù	ù
Ú	ú
Û Ũ Ū Ŭ	û ũ ū ŭ
Ü	ü
Ů	ů
Ý	ý
Ÿ	ÿ
Ž	ž

COMPARE
<
≤
=
≠
≥
>
[
]
{
}

PUNCT
,
“
#
&
‘
*
:
;
@
\
_
|
~

TABLE OF CONSTANTS

This lists all constants contained in the catalogue CONST. Names printed in a golden field represent fundamental or measured constants, while the other ones may be derived from them. The constants π and **e** are also found on the keyboard directly. The characters necessary to get to a specific function in the catalogue are printed bold in this index – each red character must be replaced by a ▼ .

Name	Value	Dimension	Remarks
a	365,2425	d	Gregorian year (per definition)
a_0	5,291772083E-11	m	Bohr radius $= \alpha / 4\pi R_\infty$
atm	1,01325E5	Pa/atm	Standard pressure p_0 (per definition)
c	2,99792458E8	m/s	Vacuum speed of light (per definition)
c_1	3,7417712E-16	$m^2 \cdot W$	First radiation constant $= 2\pi \cdot h \cdot c^2$
c_2	0,014387752	$m \cdot K$	Second radiation constant $= hc/k$
e	2,718281828459045	1	
eV	1,602176462E-19	J	= Electron charge * 1V. Remember J = V A s .
F	96485,3415	$\frac{A \cdot s}{mol}$	Faraday's constant = e N_A
g	9,80665	m/s^2	Standard earth acceleration (per definition)
G	6,6742867E-11	$\frac{m^3}{kg \cdot s^2}$	Newton's gravitation constant
g_e	2,002319304362	1	Landé's g-factor
G_o	7,748091696E-5	$1/\Omega$	Conductance quantum $= 2e^2/h$
h	6,62606876E-34	J s	Planck constant
$\hbar$	1,054571596E-34		$= h/2\pi$
k	1,3806503E-23	J/K	Boltzmann constant $= R/N_A$
m_e	9,10938188E-31	kg	Electron mass

Name	Value	Dimen-sion	Remarks
m_n	1,67492716E-27	kg	Neutron mass
m_p	1,67262158E-27		Proton mass
m_u	1,66053873E-27		Atomic unit mass = 10^{-3}kg / N_A
m_μ	1,88353109E-28		Myon mass
N_A	6,02214199E23	$1/mol$	Avogadro's number
R	8,314472	$\frac{J}{mol \cdot K}$	Molar gas constant
r_e	2,817940285E-15	m	Classical electron radius $= \alpha^2 \cdot a_0$
R_K	25812,80756	Ω	Von Klitzing constant $= h/e^2$
R_∞	10973731,5685	$1/m$	Rydberg constant $= \alpha^2 m_e c/2h$
T_o	273,15	K	= 0°C, standard temperature (per definition)
t_p	5.39124E-44	s	Planck time $= \sqrt{\hbar G/c^5}$
V_m	0,022413996	m^3/mol	Molar volume of ideal gas at standard conditions $= RT_0/p_0$
Z_o	376,730313461	Ω	Charact. impedance of vacuum $= \sqrt{\mu_0/\varepsilon_0} = \mu_0 c$
α	7,297352533E-3	1	Fine-structure constant $= e^2/4\pi\varepsilon_0\hbar c$
γ_{EM}	0,57721566490153286	1	Euler-Mascheroni constant
γ_p	267522212	$\frac{1}{s \cdot T}$	Proton gyromagnetic ratio $= 2\mu_P/\hbar$
ε_o	8,854187817E-12	$\frac{A \cdot s}{V \cdot m}$	Electric constant, vacuum permittivity $= 1/\mu_0 c^2$

Name	Value	Dimen-sion	Remarks
λ_C	2,426310215E-12	m	Compton wavelengths of electron $= h/m_e c$, neutron $= h/m_n c$, and proton $= h/m_p c$, respectively.
λ_{cn}	1,319590898E-15		
λ_{cp}	1,321409847E-15		
μ_B	9,27400899E-24	J/T	Bohr's magneton $= e\hbar/2m_e$
μ_e	-9,28476362E-24		Electron magnetic moment
μ_u	5,05078317E-27		Nuclear magneton $= e\hbar/2m_p$
μ_n	-9,662364E-27		Neutron magnetic moment
μ_p	1,410606633E-26		Proton magnetic moment
μ_o	1,2566370614E-6	N/A^2	Magnetic constant, vacuum permeability $= 4\pi \cdot 10^{-7}$ (per definition)
μ_μ	-4,49044813E-26	J/T	Müon magnetic moment
π	3, 141592653589793	1	
σ_B	5,6704E-8	$\frac{W}{m^2 \cdot K^4}$	Stefan Boltzmann constant $= \frac{2\pi^5 k^4}{15h^3c^2}$
Φ	1,61803398874989485	1	Golden ratio $= \frac{1+\sqrt{5}}{2}$
Φ_o	2,067833636E-15	V s	Magnetic flux quantum $= h/2e$
∞		1	Infinity

TABLE OF CONVERSIONS

These are the conversions contained in the new catalogue CONV. The characters necessary to get to a specific conversion in the catalogue are printed bold in this index – each red character must be replaced by a ▼ . The constant T_o may be useful for conversions, too – it is found in the catalogue CONST.

Conversion		Remarks	Class
acres→ha	* 0,4046873	Remember 1 ha = 10^4 m²	Area
atm→Pa	* 1,01325E5	Exactly	Pressure
au→km	* 1,495979E8	Astronomic units	Length
bar→Pa	* 1E5	Exactly	Pressure
bhp→W	* 745,6999	British horse power	Power
Btu→J	* 1055,056		Energy
cal→J	* 4,1868	Exactly	Energy
cm→inch	/ 2,54	Exactly	Length
feet→m	* 0,3048	Exactly	Length
flozUK→ml	* 28,41306	Remember 1 m^3 = 10^3 *l*	Volume
flozUS→ml	* 29,57353		Volume
g→oz	/ 28,34952		Mass
g→tr oz	/ 31,10348		Mass
galUK→l	* 4,54609		Volume
galUS→l	* 3,785418		Volume
ha→acres	/ 0,4046873		Area
HP$_e$→W	* 746	Exactly	Power
inch→cm	* 2,54	Exactly	Length
J→Btu	/ 1055,056		Energy
J→cal	/ 4,1868	Exactly	Energy
J→**k**Wh	/ 3,6E6	Exactly, since 1 h = 3600 s	Energy
kg→lbm	/ 0,4535924		Mass
km→au	/ 1,495979E8		Length
km→l.y.	/ 9,460730E12	Light years	Length
km→**m**i	/ 1,609344	Exactly	Length
km→**nm**i	/ 1,852	Nautical miles, exactly	Length

Conversion		Remarks	Class
km→pc	/ 3,085678E16	Parsec	Length
kWh→J	* 3,6E6	Exactly	Energy
l→galUK	/ 4,54609		Volume
l→galUS	/ 3,785418		Volume
lbf→N	* 4,448222		Force
lbm→kg	* 0,4535924		Mass
l.y.→km	* 9,460730E12		Length
m→feet	/ 0,3048	Exactly	Length
mi→km	* 1,609344	Exactly	Length
ml→flozUK	/ 28,41306		Volume
ml→flozUS	/ 29,57353		Volume
mmHg→Pa	* 133,3224	1 mmHg = 1 torr	Pressure
N→lbf	/ 4,448222		Force
nmi→km	* 1,852	Exactly	Length
oz→g	* 28,34952		Mass
Pa→atm	/ 1,01325E5	Exactly	Pressure
Pa→bar	/ 1E5	Exactly	Pressure
Pa→mmHg	/ 133,3224		Pressure
pc→km	* 3,085678E16		Length
PS(hp)→W	* 735,4988		Power
sh ton→t	* 0,9071847		Mass
t→sh ton	/ 0,9071847		Mass
t→ton	/ 1,016047		Mass
ton→t	* 1,016047		Mass
tr oz→g	* 31,10348		Mass
W→bhp	/ 745,6999		Power
W→HP_e	/ 746	Exactly	Power
W→PS(hp)	* 735,4988		Power
°C→°F			Temperature
°F→°C			Temperature

ERROR MESSAGES

Depending on error conditions, the following messages will be displayed in the dot matrix, while the part following the red slash will show up in the numeric display.

Message	Explanation and Examples
bad date / error	Invalid date format or incorrect date in input.
bad digit / error	Invalid digit in input, e.g. 9 in octal, +/- in unsigned mode or F in FLOAT.
bad mode / error	Calling an operation in a mode where it is not defined, e.g. SIN in hexa-decimal.
no such / label	Attempt to address an undefined label.
out of range / error	• An address exceeds the valid domain. • An argument is out of the valid domain for this operation. May also occur in indirect addressing. • A block register operation (e.g. R-COPY) attempts to go out of valid register numbers (0 … 99).
SLV ∫ Σ Π / nested	Nested use of solve, integrate, sum or product is not allowed.
too long / error	Input exceeding the length of the command line.
undefined / op-code	An instruction with an undefined op-code occurred.
undefined / result	• ASIN(x) or ACOS(x) for |x| > 1. • Roots of numbers < 0. • Logarithms with arguments ≤ 0. • tan(90°) and equivalent. • 0^0. • Γ(0). Some of these operations may be executed if preceded by the key CPX .
word size / error	Stack content is too big for selected word size.
∞ / error	• Division of a positive number by zero. • Divergent sum or product or integral. • Positive overflow in FLOAT.
– ∞ / error	• Division of a negative number by zero. • Divergent sum or product or integral. • Negative overflow in FLOAT.
>8 levels / nested	Subroutine nesting exceeds 8 levels.

Any key will wipe out the error message displayed and execute with the stack contents present.

Edition	Date	Release notes
1	9.12.08	Start
1.1	15.12.08	Added the table of indicators; added NAND, NOR, XNOR, RCLWS, STOWS, //, N, SERR, SIGMA, < and >; deleted HR, INPUT, 2 flag commands, and 2 conversions; extended explanations for addressing and COMPLEX & …; put XOR on the keyboard; corrected errors.
1.2	4.1.09	Added ASRN, CBC?, CBS?, CCB, SCB, FLOAT, MIRROR, SLN, SRN, >BIN, >DEC, >HEX, >OCT, BETA, D>R, DATE, DDAYS, D.MY, M.DY, Y.MD, CEIL, FLOOR, DSZ, ISZ, D>R, R>D, EMGAM, GSB, LNBETA, LNGAMMA, MAX, MIN, NOP, REAL, RJ, W and WINV, ZETA, %+ and %-; renamed the top left keys B, C, and D, and bottom left EXIT.
1.3	17.1.09	Added AIP, ALENG, ARCL, AROT, ASHF, ASTO, ATOX, XTOA, AVIEW, CLA, PROMPT (all taken from 42S), CAPP, FC?C, FS?C, SGMNT, and the …# commands; renamed NBITS to BITS and STOWS to WSIZE; specified the bit commands closer; deleted the 4 carry bit operations.
1.4	10.2.09	Added CONST and a table of constants provided, D>J and J>D, LEAP?, %T, RCL and STO ▲ and ▼, and 2 forgotten statistics registers; deleted CHS, EMGAM, GSB, REAL and ZETA; purged and renamed the bit operations; renamed many commands.
1.5	5.3.09	Added RNDINT, CONV and its table, a memory table, the description of XEQ B, C, D to the operation index, and a and g_e to the table of constants; put CLSTK on a key, moved CLΣ and FILL, changed the % and log labels on the keyboard, put CLALL in X.FCN; checked and cleaned alpha mode keyboard and added a temporary alpha keyboard; rearranged the alphabet to put Greek after Latin, symbols after Greek consistently; separated the input and non programmable commands; cleaned the addressing tables.
1.6	12.8.09	Added BASE, DAYS+, DROP, DROPY, E3OFF, E3ON, FC?F, FC?S, FIB, FS?F, FS?S, GCD, LCM, SETDAT, SETTIM, SET24, SINC, TIME, VERS, αDAY, αMONTH, αRC#; %Σ, as well as F-, t-, and χ^2-distributions and their inverses; reassigned DATE, modified DENMAX, FLOAT, αROT, and αSHIFT; deleted BASE arithmetic, BIN, DEC, HEX, and OCT; updated the alpha keyboards; added flags in the memory table; included indirect addressing for comparisons; added a paragraph about the display; updated the table of indicators; corrected errors.
1.7	9.9.09	Added P.FCN and STAT catalogues, 4 more conversions, 3 more flags, Greek character access, CLFLAG, DECOMP, DENANY, DENFAC, DENFIX, Iβ, IΓ, αDATE, αRL, αRR, αSL, αSR, αTIME, 12h, 24h, fraction mode limits, normal distribution and its inverse for arbitrary μ and σ, and Boolean operations working within FLOAT; deleted αROT and αSHIFT, the timer, and forced radians after inverse hyperbolics; renamed WINV to W^{-1}, and beta and gamma commands to Greek; added tables of catalogue contents; modified label addressing; relabeled PRGM to P/R and PAUSE to PSE; swapped SHOW and PSE as well as Δ% and % on the keyboard; relabeled Q; corrected CEIL and FLOOR; updated X.FCN and alpha commands; updated the virtual alpha keyboard.
1.8	29.10.09	Added R-CLR, R-COPY, R-SORT, R-SWAP, RCLM, STOM, alpha catalogues, 1 more constant and some more conversions, a table of error messages, as well as the binomial, Poisson, geometric, Weibull and exponential distributions and their inverses; renamed some commands; put SQRT instead of π on hotkey D.
1.9	14.12.09	Added two complex comparisons; swapped and changed labels in the top three rows of keys, dropped CLST; completed function descriptions in the index.