

C47 Full names				
FullName	Type	Label	Catalog	Description
$(-1)^x$	Function (monadic)	$(-1)^x$	$(-1)^x$	Cosine of $\pi x$ [+ i * sine of $\pi x$ for complex parameter] (Esp. for non-integer x)
(Data) entry?	Command	ENTRY?	ENTRY?	Test internal entry flag for data entry
(Plot) Central	Command (special)	CENTRL	CENTRL	Orthogonal curve fit and show parameters, requires 30 data pairs (Opens the graph mode and (re)starts menu SCATR)
$[\Sigma+]$ <empty>	Setting	->nil		Assign <empty> to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [<empty>] (inverse))
$[\Sigma+]$ <f>	Setting	->f		Assign <f> to $[\Sigma+]$ key for NORMAL as well as USER mode ; disables longpress on $[\Sigma+]$ and the use of this key as A in ASM 2-character search (Shown in KEYMAP as [g] (inverse) ; also see Ref : Assignment for regular assignment of shift functions)
$[\Sigma+]$ <g>	Setting	->g		Assign <g> to $[\Sigma+]$ key for NORMAL as well as USER mode ; disables longpress on $[\Sigma+]$ and the use of this key as A in ASM 2-character search (Shown in KEYMAP as [g] (inverse) ; also see Ref : Assignment for regular assignment of shift functions)
$[\Sigma+]$ a <sup>b/c</sup>	Setting	->a <sup>b/c</sup>		Assign a <sup>b/c</sup> to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [a <sup>b/c</sup> ] (inverse))
$[\Sigma+]$ CC	Setting	->CC		Assign CC to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [CC] (inverse))
$[\Sigma+]$ DRG	Setting	->DRG		Assign DRG to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [DRG] (inverse))
$[\Sigma+]$ f/g	Setting	->f/g		Assign f/g to $[\Sigma+]$ key for NORMAL as well as USER mode ; disables longpress on $[\Sigma+]$ and the use of this key as A in ASM 2-character search (Shown in KEYMAP as [g] (inverse) ; also see Ref : Assignment for regular assignment of shift functions)
$[\Sigma+]$ HOME	Setting	->HOME		Assign HOME to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [HOME] (inverse))
$[\Sigma+]$ i	Setting	->i		Assign i to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [i] (inverse))
$[\Sigma+]$ MyMenu	Setting	->MyM		Assign MyMenu to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [MyMenu] (inverse))
$[\Sigma+]$ PRGM	Setting	->PRGM		Assign PRGM to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [PRGM] (inverse))
$[\Sigma+]$ SNAP	Setting	->SNAP		Assign SNAP to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [SNAP] (inverse))
$[\Sigma+]$ USER	Setting	->USER		Assign USER to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [USER] (inverse))
$[\Sigma+]$ $\alpha$	Setting	-> $\alpha$		Assign $\alpha$ to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as [ $\alpha$ ] (inverse))
$[\Sigma+]$ $\Sigma+$	Setting	-> $\Sigma+$		Assign $\Sigma+$ to $[\Sigma+]$ key for NORMAL mode ONLY & deactivate USER mode (Shown in KEYMAP as $\Sigma+$ (normal))
$\sqrt{(1+x^2)}$	Function (legacy ; monadic)	$\sqrt{(1+x^2)}$	$\sqrt{(1+x^2)}$	$\sqrt{(1+x^2)}$
$^{\circ}\text{C}$ to $^{\circ}\text{F}$	Function (linked ; monadic)	$^{\circ}\text{C} \rightarrow ^{\circ}\text{F}$		Convert degrees Celsius to degrees Fahrenheit ( $^{\circ}\text{F}$ )
$^{\circ}\text{F}$ to $^{\circ}\text{C}$	Function (linked ; monadic)	$^{\circ}\text{F} \rightarrow ^{\circ}\text{C}$		Convert degrees Fahrenheit to degrees Celsius ( $^{\circ}\text{C}$ )
1	Button (black)	1		Soft button 1 for selection by KEYG or KEYX function
10	Button (black)	10		Soft button 10 for selection by KEYG or KEYX function
10 to the power x	Function (monadic)	$10^x$	$10^x$	Raise 10 to the power in the X-register
10s complement	Setting (pgm)	10CMPL		Set BCD 10s complement mode for shortint
11	Button (black)	11		Soft button 11 for selection by KEYG or KEYX function
12	Button (black)	12		Soft button 12 for selection by KEYG or KEYX function
12 hour clock	Setting (pgm)	CLK12	CLK12 ( )	12 hour time display (HH:MM ; HH:MMam/pm)

FullName	Type	Label	Catalog	Description
13	Button (black)	13		Soft button 13 for selection by KEYG or KEYX function
14	Button (black)	14		Soft button 14 for selection by KEYG or KEYX function
15	Button (black)	15		Soft button 15 for selection by KEYG or KEYX function
16	Button (black)	16		Soft button 16 for selection by KEYG or KEYX function
17	Button (black)	17		Soft button 17 for selection by KEYG or KEYX function
18	Button (black)	18		Soft button 18 for selection by KEYG or KEYX function
1's complement	Setting (pgm)	1COMPL	1COMPL ( )	Set 1's complement mode for shortint (SBI depends on SBint)
1st derivative	Function (monadic)	Calc f'	Calc f'	Calculate value of 1st derivative for (starred) variable (f' =)
1st eccentricity squared	Constant (#47)	Se <sup>2</sup>		sq.eccent1 Se <sup>2</sup> = +6.69437999014 × 10 <sup>-3</sup>
1st radiation constant	Constant (#05)	c <sub>1</sub>		c.radiatn1 c <sub>1</sub> = +3.741771852192758011367155555929985 × 10 <sup>-16</sup> (Wm <sup>2</sup> )
2	Button (black)	2		Soft button 2 for selection by KEYG or KEYX function
2 to the power x	Function (monadic)	2 <sup>x</sup>	2 <sup>x</sup>	Raise 2 to the power in the X-register
24 hour clock	Setting (pgm)	CLK24	CLK24 (•)	24 hour time display (HH24:MM ; HH:MMam/pm)
2nd derivative	Function (monadic)	Calc f''	Calc f''	Calculate value of 2nd derivative for (starred) variable (f'' =)
2nd eccentricity squared	Constant (#48)	Se' <sup>2</sup>		sq.eccent2 Se' <sup>2</sup> = +6.73949674228 × 10 <sup>-3</sup>
2nd radiation constant	Constant (#06)	c <sub>2</sub>		c.radiatn2 c <sub>2</sub> = +1.438776877503933802146671601543912 × 10 <sup>-2</sup> (mK)
2's complement	Setting (pgm)	2COMPL	2COMPL (•)	Set 2's complement mode for shortint (SBI depends on SBint)
3	Button (black)	3		Soft button 3 for selection by KEYG or KEYX function
4	Button (black)	4		Soft button 4 for selection by KEYG or KEYX function
5	Button (black)	5		Soft button 5 for selection by KEYG or KEYX function
6	Button (black)	6		Soft button 6 for selection by KEYG or KEYX function
7	Button (black)	7		Soft button 7 for selection by KEYG or KEYX function
8	Button (black)	8		Soft button 8 for selection by KEYG or KEYX function
9	Button (black)	9		Soft button 9 for selection by KEYG or KEYX function
9s complement	Setting (pgm)	9CMPL		Set BCD 9s complement mode for shortint
A	Character	À		Character À (65)
A acute	Character	Á		Character Á (193)
a acute lowercase	Character	á		Character á (225)
A breve	Character	Â		Character Â (258)
a breve lowercase	Character	ã		Character ã (259)
A circumflex	Character	Ã		Character Ã (194)
a circumflex lowercase	Character	ä		Character ä (226)
A diaeresis	Character	Ä		Character Ä (196)
a diaeresis lowercase	Character	ä		Character ä (228)
A grave	Character	À		Character À (192)
a grave lowercase	Character	à		Character à (224)
A lowercase	Character	a		Character a (97)
A macron	Character	Ā		Character Ā (256)
a macron lowercase	Character	ā		Character ā (257)
A ogonek	Character	Ą		Character Ą (260)
a ogonek lowercase	Character	ą		Character ą (261)
A ring	Character	Å		Character Å (197)
a ring lowercase	Character	å		Character å (229)
A tilde	Character	Ã		Character Ã (195)
a tilde lowercase	Character	ã		Character ã (227)
Absolute	Function (legacy)	ABS	ABS	Magnitude (absolute value) of complex number (Superseded by  x  (keyboard f[510]))
Access blue functions	MENU	BLUE47	BLUE47	Access all C47 g-shift keyboard and a few f-shift keyboard functions, supporting layout DM42 (Assigned to calculator function SETUP (f[0]) in layout DM42)
Access DMCP menu	Command (HW)	DMCP	DMCP	Access DMCP menu ; hardware only ; use EXIT to return without reset
Accuracy	Variable (real)	ACC	ACC	Accuracy of the integrator (reserved real variable) (ACC :)

FullName	Type	Label	Catalog	Description
acre <sub>us</sub> to ha	Function (linked ; monadic)	acre <sub>us</sub> → ha		Convert US acre to hectare (ha:)
acre to ha	Function (linked ; monadic)	acre→ha		Convert acre to hectare (ha:)
Activate USB disk	Command (HW)	ActUSB	ActUSB	Activate USB disk without exiting to DMCP ; hardware only
Add	Function (dyadic)	+	+	Add X to Y ; concatenate X and Y (Concatenation includes numbers, dates and times + strings and vice versa)
Add column	Command	COL+1	M.COL+1	Add column to matrix, at the right edge
Add differential curve	Setting	Δy/Δx		Add another curve, the graphical point-to-point differential of the main graph
Add integral curve	Setting	ΣyΔx		Add another curve, the graphical integral calculated from the point-to-point discrete integral (Vertical placement of the integral is at y = RMS(y))
Add into	Function (monadic)	STO+	STO+	Add into (STO+ __ TamStoRcl menu)
Add lap time to STATS	Command	LAP→Σ		Add lap time to STATS (statistics matrix) and reset lap timer ; subscript shows number of data points ([+] (STOPW active))
Add margin to cost	Function (dyadic)	%+MG	%+MG	Add margin of X to cost of Y
Add RMS curve	Setting	RMS		Add another curve, the graphical root mean square, accumulated from left to right.
Add row	Command	ROW+1	M.ROW+1	Add row to matrix, at the bottom edge
Add time to STATS	Command	TIM→Σ		Add time to STATS (statistics matrix) ; subscript shows number of data points ([Σ+] (STOPW active))
Advanced	MENU	ADV	ADV	Advanced functions
AE	Character	Æ		Character Æ (198)
ae lowercase	Character	æ		Character æ (230)
A-F keys	Setting	KEY <sub>A-F</sub>	KEY <sub>A-F</sub> [•]	Set row 2 keys to be used for entry of digits A-F while in #BASE mode
All digits notation	Setting (pgm)	ALL	ALL <sub>3</sub> (•)	Set numeric display mode to ALL digits notation ; display all digits as far as possible with max. nn decimal zeros (ALL __ TamNonReg menu)
All variables	MENU (ASM)	ALL	ALLVARS	Auto-generated catalog of all variables
ALL/FIX ENG(ineering)	Flag	ALLENG		Real number exceeding displayable range for ALL or FIX will be shown as either ENG (if set) or SCI (if clear)
Allocate local registers	Command (PEM)	LocR	LocR	Allocate n local registers and 16 local flags (available in the current routine) (LocR __ TamNonReg menu)
AllOp's	User Program	AllOp's		AllOp's test program, preloaded from testPgms.bin
Almost equal	Character	≈		Character ≈ (8776)
ALPHA	Character	A		Character A (913)
Alpha capslock	Flag (system)	αCAP		Set for capital letters, clear for lower case
Alpha Entry	Flag	ALP.IN		Alpha entry active
Alpha graphics	Command	AGRAPH	AGRAPH	Show a 64 column pattern according to variable GRAMOD (OR: 0 ; SET: 1 ; OFF: 2 ; XOR: 3) (AGRAPH __ Tam menu)
Alpha input	MENU (item)	α		Alpha menu is used to enter, edit and clear alpha input (long [XEQ] ; Info : AIM = Alpha Input Mode ; opens menu α in UPPERCASE)
Alpha input	MENU	ALPHA	ALPHA	Alpha menu is used to enter, edit and clear alpha input
Alpha input mode	Flag	ALPHA		Set for AIM, else clear
Alpha international	MENU (ASM)	αINTL	αINTL	Uppercase international characters (Lowercase menu via [▼] ; Type characters 1-2 to search)
alpha international	MENU (ASM)	αintl lower	αintl	Lowercase international characters (Uppercase menu via [▲] ; Type characters 1-2 to search)
Alpha length?	Function (monadic)	αLENG?	αLENG?	Get string length (αLENG? __ Tam menu)
Alpha lock downwards	Alpha-shift	CASE DN		Move down alpha lock from N to A to a (alpha f [▼])
Alpha lock upwards	Alpha-shift	CASE UP		Move up alpha lock from a to A to N (alpha f [▲])
alpha lowercase	Character	α		Character α (alpha g[XEQ] ; Code : 945)
Alpha Math	MENU	αMath	αMath	Mathematical symbols
Alpha miscellaneous	MENU	αMisc	αMisc	Special characters
Alpha parse	Command (strike)	αPARSE	αPARSE	Parse alpha input for numeric content (in development)

FullName	Type	Label	Catalog	Description
Alpha position?	Function (monadic)	$\alpha$ POS?	$\alpha$ POS?	Get position in string in variable or stack of substring in X (starting from 0) ( $\alpha$ POS? __ Tam menu)
Alpha rotate left	Function (monadic)	$\alpha$ RL	$\alpha$ RL	Alpha rotate left ( $\alpha$ RL __ Tam menu)
Alpha rotate right	Function (monadic)	$\alpha$ RR	$\alpha$ RR	Alpha rotate right ( $\alpha$ RR __ Tam menu)
Alpha shift left	Function (monadic)	$\alpha$ SL	$\alpha$ SL	Alpha shift left ( $\alpha$ SL __ Tam menu)
Alpha shift right	Function (monadic)	$\alpha$ SR	$\alpha$ SR	Alpha shift right ( $\alpha$ SR __ Tam menu)
Alpha string	MENU	$\alpha$ .FN	$\alpha$ .FN	Alpha (string) functions
Alpha to X	Function (monadic)	$\alpha \rightarrow x$	$\alpha \rightarrow x$	Convert first alpha character of string in variable or register to character code (hexadecimal) ( $\alpha \rightarrow x$ __ Tam menu)
alpha tonos lowercase	Character	á		Character á (940)
ALPHA..OMEGA	MENU	A..Ω	A..Ω	Uppercase Greek characters (Lowercase menu via [▼])
alpha..omega	MENU	$\alpha$ ..ω lower	$\alpha$ ..ω	Lowercase Greek characters (Uppercase menu via [▲])
Amortisation	MENU (strike)	AMORT	AMORT	Amortisation
Ampersand	Character	&		Character & (38)
And	Character	∧		Character ∧ (8743)
AND	Function (dyadic)	AND	AND	Logical AND (bitwise)
Angle	Character	∠		Character ∠ (8736)
Angle conversion	MENU	Angle:	Angle:	Convert between units of angle
Angle variables	MENU	ANGLES	ANGLES	Auto-generated catalog of variables of the specified type : angle
Angular display mode	Variable (longint)	ADM	ADM	Angular display mode (system real variable, write protected) (RAD: 2 ; ; GRAD: 4 ; DEG: 0 ; [D.MS: 1 ; MULπ: 3])
Annual interest rate	Variable (real)	I/a	I%/a	Annual interest rate as percentage (reserved real variable ; Use 10 for 10% not 0.1) (I%/a = I%YR = APR = )
Anti-clockwise vectors	Setting	VECT		PLOTMNU only : treat coordinate pairs as vectors, anti-clockwise from the positive x-axis as reference
Apostrophe	Character	'		Character ' (39)
Arc cosiné	Function (legacy)	ARCCOS	ARCCOS	Inverse cosine (Superceded by: ARCCOS (keyboard: f[COS]))
Arc cosine	Function (monadic)	ACOS	ACOS	Inverse cosine
Arc sine	Function (monadic)	ASIN	ASIN	Inverse sine
Arc sine	Function (legacy)	ARCSIN	ARCSIN	Inverse sine (Superceded by: ARCSIN (keyboard: f[SIN]))
Arc tangent	Function (monadic)	ATAN	ATAN	Inverse tangent
Arc tangent	Function (legacy)	ARCTAN	ARCTAN	Inverse tangent (Superceded by: ARCTAN (keyboard: f[TAN]))
Area conversion	MENU	Area:	Area:	Convert between units of area
Argument (angle)	Function (monadic)	∠	∠	Argument (angle) of complex number
Arithmetic Geometric Mean	Function (dyadic)	AGM	AGM	Arithmetic geometric mean of X and Y
Arithmetic means	Command	$\bar{x}$	$\bar{x}$	Arithmetic means ( $\bar{x}$ ; $\bar{y}$ = (2 stack levels))
Arithmetic shift right	Function (monadic)	ASR	ASR	Arithmetic shift right by n (ASR __ TamNonReg menu)
Assess	MENU (item)	ASSESS		Display graphical assessments of the selected regression curves
Assess	MENU	ASSESS		Display graphical assessments of the selected regression curves
Assign	Command	ASN	ASSIGN	Assign function, menu or character to a keyboard key or to a button in a menu (see Ref : Assignment for options including creating user defined menus) (Use button ASN ([α.1] fF6) for character assignments ; this button is not shown in Program Entry Mode)
Asterisk	Character	*		Character * (42)
Asymptotically equal to	Character	≈		Character ≈ (8771)
At	Character	@		Character @ (64)
atm to Pa	Function (linked ; monadic)	atm→Pa		Convert atmosphere to Pascal (Pa:)
Atomic mass constant	Constant (#30)	$m_u$		mass.atom $m_u = +1.6605390666 \times 10^{-27}$ (kg)
au to m	Function (linked ; monadic)	au→m		Convert astronomical unit to meter (m:)
Auto Execute (Program)	Flag	AUTXEQ		Auto-execute program when turning on calculator

FullName	Type	Label	Catalog	Description
Auto Off (Calculator)	Flag	AUTOFF		Automatic shutdown active (10 m)
Automatic Stack Lift	Flag (system)	ASLIFT		Automatic stack lift active (SBI is SIM only)
Autoscale x-axis	Setting	X.AXIS		PLOTMNU only : autoscale Y so that the x-axis becomes visible
Autoscale y-axis	Setting	Y.AXIS		PLOTMNU only : autoscale X so that the x-axis becomes visible
Avogadro's number	Constant (#35)	$N_A$		nr.avogadro $N_A = +6.02214076 \times 10^{23}$ (/mol)
B	Character	B		Character B (66)
b lowercase	Character	b		Character b (98)
Back	Command (PEM)	BACK	BACK	Jumps n steps back in program (BACK ___ TamNonReg menu)
Back slash	Character	\		Character \ (92)
Backbone exponent	Constant (#79)	$\xi_B$		exp.bbone $\xi_B = +3.566668367128895828373073810012663 \times 10^{-1}$
Backspace	Command (nonpgm)	↵		Backspace (Clear NIM) (BKSPC)
Bairsto	User Program	Bairsto		Bairsto test program, preloaded from testPgms.bin
Bar	Symbol			Open/close absolute value (alpha g[ST0] ; Code : 124)
bar to Pa	Function (linked ; monadic)	bar→Pa		Convert bar to Pascal (Pa)
barrel to m <sup>3</sup>	Function (linked ; monadic)	barrel→m <sup>3</sup>		Convert barrel to cubic meter (m <sup>3</sup> )
Battery	Character	🔋		Character 🔋 (9258)
Battery depletion	Character	🔋		Character 🔋 (9269)
Battery voltage?	Command	BATT?	BATT?	Get battery voltage ±1 mV (V = )
BCD unsigned	Setting (pgm)	BCDUNS		Set BCD unsigned mode for shortint
Beep	Command	BEEP	BEEP	Play beep sound (4 tones)
Best Fit	Setting	BestF	BestF	Select curve fit model(s) by setting parameter value directly (BestF ___ TamNonReg menu)
Best Fit Setting?	Command	BestF?	BestF?	Get parameter for which models are selected in the LR analysis (Set in menu MODEL)
BETA	Character	B		Character B (914)
beta lowercase	Character	β		Character β (946)
Binary	Setting (pgm)	BIN	BIN	Convert X to binary and toggle binary mode (#TAM [1/x] (B))
Binary 0	Character	0		Character 0 (8718)
Binary 1	Character	1		Character 1 (8231)
Binary coded decimal	Setting	BCD	BCD [ ]	Set BCD mode for shortint
Binary logarithm	Function (monadic)	LB	LB	Binary logarithm (base 2)
Binary prefixes	Setting	1024 <sup>n</sup>	1024 <sup>n</sup> [ ]	Activate binary prefixes (powers of 1024) for UNIT mode
Binary prefixes	Flag	1024 <sup>n</sup>		Activate binary prefixes (powers of 1024) for UNIT mode
Binomial & Negative binomial distribution	MENU	Binom:	Binom:	Binomial & Negative binomial probability distribution (discrete) (RegP = p = probability ; RegN = n = number of samples)
Binomial (quantile)	Function (monadic)	Binom <sup>-1</sup>	Binom <sup>-1</sup>	Binomial inverse cumulative distribution
Binomial cdf (lower)	Function (monadic)	Binom <sub>⋄</sub>	Binom <sub>⋄</sub>	Binomial cumulative distribution (lower tail)
Binomial cdf (upper)	Function (monadic)	Binom <sup>⋄</sup>	Binom <sup>⋄</sup>	Binomial cumulative distribution (upper tail)
Binomial pmf	Function (monadic)	Binom <sub>p</sub>	Binom <sub>p</sub>	Binomial probability mass function
Bit clear?	Function (monadic)	BC?	BC?	Test if bit n is clear (BC? __ TamNonReg menu)
Bit set?	Function (monadic)	BS?	BS?	Test if bit n is set (BS? __ TamNonReg menu)
Bits	MENU	BITS	BITS	Bitwise operations
B <sub>n</sub>	Function (monadic)	B <sub>n</sub>	B <sub>n</sub>	Bernoulli number (new definition)
B <sub>n</sub> *	Function (monadic)	B <sub>n</sub> *	B <sub>n</sub> *	Bernoulli number (old definition)
Bohr radius	Constant (#01)	a <sub>0</sub>		rad.bohr a <sub>0</sub> = +5.29177210903 × 10 <sup>-11</sup> (m)
Bohr's magneton	Constant (#65)	μ <sub>B</sub>		magn.both μ <sub>B</sub> = +9.274010078 × 10 <sup>-24</sup> (J/T)
Boltzmann constant	Constant (#20)	k <sub>B</sub>		c.boltzmn k <sub>B</sub> = +1.380649 × 10 <sup>-23</sup> (J/K)
Box markers	Setting (pgm)	BOX		Use boxes as markers for the main graph
brds to in.	Function (linked ; monadic)	brds→in.		Convert beardsecond to inch (in.)
brds to m	Function (linked ; monadic)	brds→m		Convert beardsecond to meter (m.)
BST	Command	▲		Scroll Up Menu (or SHOW) or Back Step
BST	Character	Δ		Character Δ (8649)

FullName	Type	Label	Catalog	Description
BST	Character	Δ		Character Δ (8649)
Btu to J	Function (linked ; monadic)	Btu→J		Convert British Thermal Unit to Joule (J:)
Bullet	Character	•		Character • (8729)
Buzzer	Command (HW)	BUZZ	BUZZ	Play sound (input from stack : frequency in Hz (0 = silent) ; duration in ms (max 2000)) ; hardware only
Byte swap	Function (monadic)	B.SWP	B.SWP	Swap bytes
C	Character	C		Character C (67)
C acute	Character	Ć		Character Ć (262)
c acute lowercase	Character	ć		Character ć (263)
C caron	Character	Č		Character Č (268)
c caron lowercase	Character	č		Character č (269)
C cedilla	Character	Ç		Character Ç (199)
c cedilla lowercase	Character	ç		Character ç (231)
c lowercase	Character	c		Character c (99)
C47 layout	Layout	C47		C47 keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS-R) (C47: Classic single shift (DM42))
cal to J	Function (linked ; monadic)	cal→J		Convert calorie to Joule (J:)
Calculate	Command (nonpgm)	Calc f	Calc f	Calculate value of the result for the expression (f =)
Capslock	Setting	CAPS		Lock uppercase alpha input (Toggling clears Numlock ; also sets initial case of menus A..Ω/α..ω and αINTL/αintl)
carat to g	Function (linked ; monadic)	carat→g		Convert carat to gram (g:)
Carry	Flag	CARRY		Status of carry bit (SBI depends on SBoc)
Case	Command (PEM)	CASE	CASE	Case statement, skips number of steps given by chosen variable (normally followed by number of GTO statements) (CASE __ Tam menu)
Cash flow	MENU (strike)	<del>CASHFL</del>	<del>CASHFL</del>	Cash flow
Catalan's constant	Constant (#14)	G <sub>c</sub>		c.catalan G <sub>c</sub> = +9.159655941772190150546035149323841 × 10 <sup>-1</sup>
Catalog	MENU	CAT		Catalog of all items (functions, characters, programs, variables, menus)
Cauchy (quantile)	Function (monadic)	Cauch <sup>-1</sup>	Cauch <sup>-1</sup>	Cauchy inverse cumulative distribution
Cauchy cdf (lower)	Function (monadic)	Cauch <sub>Δ</sub>	Cauch <sub>Δ</sub>	Cauchy cumulative distribution (lower tail)
Cauchy cdf (upper)	Function (monadic)	Cauch <sub>Δ</sub>	Cauch <sub>Δ</sub>	Cauchy cumulative distribution (upper tail)
Cauchy fit	Setting	CauchF	CauchF [ ]	Set BestF to include Cauchy curve fitting ; deselects OrthoF (Adds 128 to BestF ; TI : L.R. selected to <BestF>)
Cauchy pdf	Function (monadic)	Cauch <sub>p</sub>	Cauch <sub>p</sub>	Cauchy probability density function
Cauchy-Lorentz distribution	MENU	Cauch:	Cauch:	Cauchy-Lorentz probability distribution (continuous) (RegM = x <sub>0</sub> = location ; RegS = γ = scale)
Ceiling	Function (monadic)	ceil	ceil	Ceiling (type real)
Cent	Character	¢		Character ¢ (162)
Change sign	Function (monadic)	CHS	CHS	CHange Sign
Characteristic impedance of vacuum	Constant (#54)	Z <sub>0</sub>		imped.vac Z <sub>0</sub> = +3.767303134617706554681984004203193 × 10 <sup>2</sup> (Ω)
Characters	MENU	CHARS	CHARS	Access to all character submenus (international, greek, math, MyAlpha, alphaDot)
Chebyshev polynomials (1st)	Function (dyadic)	T <sub>n</sub>	T <sub>n</sub>	Chebyshev polynomials of the 1st kind (x = X ; n = Y)
Chebyshev polynomials (2nd)	Function (dyadic)	U <sub>n</sub>	U <sub>n</sub>	Chebyshev polynomials of the 2nd kind (x = X ; n = Y)
Check mark	Character	✓		Character ✓ (10003)
CHI	Character	Χ		Character Χ (935)
chi lowercase	Character	χ		Character χ (967)
chī to m	Function (linked ; monadic)	chī→m		Convert chī to meter (m:)
Chinese formatting	Setting (pgm)	CHINA	SETCHN	Set to Chinese regional formats (date, time, calendar, number formatting) (First Gregorian day set: 1949-10-01)
Chi-squared (quantile)	Function (monadic)	(χ <sup>2</sup> ) <sup>-1</sup>	(χ <sup>2</sup> ) <sup>-1</sup>	Chi-squared inverse cumulative distribution
Chi-squared cdf (lower)	Function (monadic)	χ <sup>2</sup> <sub>Δ</sub> (x)	χ <sup>2</sup> <sub>Δ</sub> (x)	Chi-squared cumulative distribution (lower tail)
Chi-squared cdf (upper)	Function (monadic)	χ <sup>2</sup> <sub>Δ</sub> (x)	χ <sup>2</sup> <sub>Δ</sub> (x)	Chi-squared cumulative distribution (upper tail)

FullName	Type	Label	Catalog	Description
Chi-squared distribution	MENU	$\chi^2$ :	$\chi^2$ :	Chi-squared probability distribution (continuous) (RegM = v = degrees of freedom)
Chi-squared pdf	Function (monadic)	$\chi^2_p(x)$	$\chi^2_p(x)$	Chi-squared probability density function
Classic conversions	Flag	CONV <sub>HP</sub>		Set for the classic (HP) conventions for the unit conversion buttons under menu CONV
Classic Rect/Polar	Setting	RP <sub>HP</sub>	RP <sub>HP</sub> [•]	Set for the classic (HP) stack conventions for →RECT and →POLAR ; Clear to follow C47 conventions of CC, COMPLEX and <i>i</i> (swapped) (Classic means X = x (Re) , Y = y (Im) for RECT ; X = r, Y = $\theta$ for POLAR)
Classic Rect/Polar	Flag	RP <sub>HP</sub>		Set for the classic (HP) stack conventions for →RECT and →POLAR ; Clear to follow C47 conventions of CC, COMPLEX and <i>i</i> (swapped) (Classic means X = x (Re) , Y = y (Im) for RECT ; X = r, Y = $\theta$ for POLAR)
Classical electron radius	Constant (#39)	$r_e$		rad.elec $r_e = +2.8179403262 \times 10^{-15}$ (m)
Clear	MENU	CLR	CLR	Clear flags, programs, registers, stacks, variables and reset calculator
Clear (program) menu	Command	CLMENU	CLMENU	Clear the programmable menu
Clear (user) flags	Command	CLFall	CLFall	Clear all user flags (numbered and lettered) (Clear all flags?)
Clear all menus	Command	CLMall	CLMall	Clear assignments to all user defined menus (Clear all user menus?)
Clear all registers	Command	CLREGS	CLREGS	Clear all registers (Clear registers?)
Clear all variables	Command	CLVall	CLVall	Clear all user variables ; clear matrices HISTO ; STATS ; Mat_A ; Mat_B ; Mat_X (Clear all user variables?)
Clear alpha	Command	CLA		Clear alphabetic input (alpha f [↔] ; alpha g [↔] ; alpha long [↔])
Clear bit	Function (monadic)	CB	CB	Clear bit n (CB __ TamNonReg menu)
Clear current (program) variables	Command	CLCVAR	CLCVAR	Clear current program variables
Clear flag	Command	CF	CF	Clear flag (CF __ TamFlag menu)
Clear graph	Command	CLGRF	CLGRF	Clear graph
Clear LCD (screen)	Command	CLLCD	CLLCD	Clear (part of) the display, depending on X and Y
Clear modes	Command	CLRMOD	CLRMOD	Clear all menus, entry, fraction and base modes (longer [EXIT] ; alpha longer [EXIT])
Clear number	Command	CLN		Clear numeric input (NIM long [↔])
Clear registers	Command	R-CLR	R-CLR	Clear registers (sss.nn means clear registers from sss through sss + nn - 1)
Clear stack	Command	CLSTK	CLSTK	Clear all stack data (long [↔])
Clear statistics	Command	CLΣ	CLΣ	Clear all statistics data ; delete matrices HISTO and STATS
Clear TVM variables	Command	CLTVM	CLTVM	Clear TVM variables
Clear X	Command	CLX	CLX	Clear X-register
Clock	MENU	CLK	CLK	Clock functions, including setting date and time and julian day numbers (astronomy)
Clockwise vectors	Setting	N.VECT		PLOTMNU only : treat coordinate pairs as navigation vectors, clockwise from the positive y-axis as reference
Colon	Symbol	:		In NEW [EQN], used to name an expression (NAME : <expression>) (alpha [0] ; Code : 58)
Colon equals	Character	:=		Character := (8788)
Combinations	Function (dyadic)	$C_{yx}$	COMB	Combinations of X out of Y
Comma	Character	,		Character , (44)
Common logarithm	Symbol	LOG		Common logarithm (base 10) (alpha g [LOG] ("LOG"))
Common logarithm	Function (monadic)	LOG	LOG	Common logarithm (base 10)
Complement	Character	⊂		Character ⊂ (8705)
Complementary error function	Function (monadic)	erfc	erfc	Complementary error function
Complete elliptic integral (1st)	Function (monadic)	K(m)	K(m)	Complete elliptic integral of the 1st kind (m = X)
Complete elliptic integral (2nd)	Function (monadic)	E(m)	E(m)	Complete elliptic integral of the 2nd kind (m = X)
Complete elliptic integral (2rd)	Function (dyadic)	Π(n,m)	Π(n,m)	Complete elliptic integral of the 3rd kind (n = X ; m = Y)
Complex	Function (dyadic)	COMPLEX	COMPLEX	Convert to or from complex number (a ENTER b COMPLEX returns a+bi or a∠b (using angle tag, regardless of POLAR, or ADM) ; COMPLEX returns Y : a, X : b)
Complex C	Character	⊂		Character ⊂ (8450)
Complex functions	MENU	CPX	CPX	Complex functions

FullName	Type	Label	Catalog	Description
Complex $j$	Flag	CPX $j$		Set for the letter $j$ representing the imaginary number ( $i$ or $j$ displayed in stack and on soft buttons)
Complex multiplier	Setting	CPXmul	CPXmul [ ]	Complex multiplier (If clear, trailing $i$ or $j$ is used)
Complex number (polar)	Command	$i_{\odot}$	op_ $i_{\odot}$	Enter complex number (polar) whether POLAR is set or not ; e.g. a $i_{\odot}$ b ENTER results in $a_{\angle}b$ (according to ADM) (In NIM, works like CC with POLAR set ; displayed according to flag CPX $j$ when in RECT mode)
Complex number (rectangular)	Command	$i$	op_ $i$	Enter complex number (rectangular) whether RECT is set or not ; e.g. a $i$ b ENTER results in $a+bi$ (In NIM, works like CC with RECT set ; displayed according to flag CPX $j$ when in RECT mode)
Complex plot	Setting (pgm)	CXPLT		Use the complex plane for the main graph
Complex results	Setting	CPXRES	CPXRES [•]	Set to allow complex results for real input ; auto-set when entering complex input (SBI depends on SBcr)
Complex results	Flag	CPXRES		Set to allow complex results for real input ; auto-set when entering complex input (SBI depends on SBcr)
Complex results	Command	CPXRES1	CPXRES1	Allow complex results for real input ; an error will not occur for such events (For programming purposes)
Complex solver (stack limits)	Command (nonpgm)	cpxSlv $\times$	cpxSlv $\times$	Solve the algebraic expression ( $= 0$ ) entered in NEW [EQN] ; interrupt by keypress ; tolerance set by SDIGS ; monitoring set by MONIT ; (uses registers R81-R98) (Result Code = ; Accuracy $\approx$ ; $\langle var \rangle_{PREV} =$ ; $\langle var \rangle =$ ; (4 stack levels))
Complex solver (variable limits)	Command (nonpgm)	cpxSlv	cpxSlv	Solve the algebraic expression ( $= 0$ ) entered in NEW [EQN] ; interrupt by keypress ; tolerance set by SDIGS ; monitoring set by MONIT ; (uses registers R81-R98) (Result Code = ; Accuracy $\approx$ ; $\langle var \rangle_{PREV} =$ ; $\langle var \rangle =$ ; (4 stack levels))
Complex to real	Function (monadic)	CX $\rightarrow$ RE	CX $\rightarrow$ RE	Convert complex to reals (in POLAR, using angle tag or ADM) (Re = ; Im = or r = ; $\theta =$ (2 stack levels))
Complex variables	MENU	CPXS	CPXS	Auto-generated catalog of variables of the specified type : complex
Complex?	Function (monadic)	CPX?	CPX?	Test X is complex
Compose 3x1 matrix	Function (triadic)	zyx $\rightarrow$ M	zyx $\rightarrow$ M	Create 3x1 matrix from ZYX (M $\rightarrow$ zyx)
Compose-cut	Command	CC	CC	Complex closing, composing, cutting, and converting (a CC b ENTER returns $a+bi$ or $a_{\angle}b$ (using angle tag, regardless of POLAR, or ADM) ; CC returns Y : a, X : b)
Compounding period	Variable (real)	cp/a	CPER/a	Compounding period per annum (reserved real variable) (CPER/a = Compounding periods p.a. = )
Compton wavelength of the electron	Constant (#61)	$\lambda_c$		wavln.elec $\lambda_c = +2.42631023867 \times 10^{-12}$ (m)
Compton wavelength of the neutron	Constant (#62)	$\lambda_{cn}$		wavln.neu $\lambda_{cn} = +1.31959090581 \times 10^{-15}$ (m)
Compton wavelength of the proton	Constant (#63)	$\lambda_{cp}$		wavln.prot $\lambda_{cp} = +1.32140985539 \times 10^{-15}$ (m)
Conductance quantum	Constant (#13)	$G_0$		cond.quant $G_0 = +7.748091729863650646680823323308764 \times 10^{-5}$ ( $\Omega^{-1}$ )
Config variables	MENU	CONFIGS	CONFIGS	Auto-generated catalog of variables of the specified type : config
Configuration	MENU (item)	CFG		Activates menu for setting system flags using FF (Flip flag) function (CAT.MENUS SYS.FL)
Confirmation	MENU	YESNO		Confirmation menu
Conjugate	Function (monadic)	conj	CONJ	Conjugate
Constant	Command	CNST	CNST	Get constant (0..78) (CNST ___ TamNonReg menu ; TI : $\langle constant \rangle$ )
Constants	MENU (ASM)	CNST	CNST	Important scientific and technical constant values (Constants preceded by "*" in programs ; Type characters 1-2 to search ; TI (temporary info) is shown in description of constants)
Contains as member	Character	$\exists$		Character $\exists$ (8715)
Contour integral	Character	$\oint$		Character $\oint$ (8750)
Convergence?	Function (dyadic)	CONVG?	CONVG?	Tests convergence of X and Y using binary coded parameter for comparison mode and special numbers ; tolerance is derived from setting SDIGS (CONVG? _ TamNonReg menu)
Convert from symmetrical components	Function (triadic)	$\rightarrow abc$	SYMtoA	Convert symmetrical components $a_0, a_1, a_2$ in Z, Y, X to 3-phase a, b, c in Z, Y, X ( $\rightarrow 012$ )
Convert to decimal hours	Function (legacy monadic)	$\rightarrow HR$	$\rightarrow HR$	Convert to decimal hours (Assumes that angle conforms to ADM ; superseded by $\rightarrow d$ [keyboard g [LOG]])



FullName	Type	Label	Catalog	Description
Convert to hours, min, sec	Function (monadic)	→h.ms	→h.ms	Convert sexagesimal format input sequence or decimal stack value to hh:mm:ss hours (NIM input treated as sexagesimal (hh.mmss) format ; stack input treated as decimal value)
Convert to symmetrical components	Function (triadic)	→ 012	AtoSYM	Convert 3-phase a, b, c in Z, Y, X to symmetrical components a0, a1, a2 in Z, Y, X (→abc)
Convert to XXX	Character	→		Character → (8658)
Convert units	MENU	CONV	CONV	Convert units (Flag CONV <sub>HP</sub> (default ON) may be set for the classic (HP) conventions for the unit conversion buttons under menu CONV)
Copy → +NRM assignment to USER	Command	COPY→U		Copy [Σ+] NORMAL assignment to [Σ+] USER assignment and activate USER mode (Does not copy →g assignment)
Copy registers	Command	R-COPY	R-COPY	Copy registers (sss.nnddd means copy registers from sss through sss + nn - 1 to registers ddd through ddd + nn - 1)
Correlation	Command	r	CORR	Correlation
Corresponds to	Character	≡		Character ≡ (8792)
Cosine	Function (monadic)	COS	COS	Cosine
Cosine	Symbol	COS		Cosine (alpha g[COS] ("COS"))
Covariance	Command	cov	COV	Covariance
Create 3x3 A-Matrix	Command	[A]	op_A	Create 3x3 A-matrix relating to Fortescue's Symmetrical Components
Cross	Function (dyadic)	x		Cross (215)
Cross (x)	Function (dyadic)	cross	CROSS	Cross product of vectors in X and Y (215)
Cross (·)	Function (dyadic)	dot	DOT	Dot product of vectors in X and Y (183)
Cross markers	Setting (pgm)	CROSS		Use crosses as markers for the main graph
Cube	Function (monadic)	x <sup>3</sup>	x <sup>3</sup>	Raise to power of 3
Cube root	Function (monadic)	$\sqrt[3]{x}$	$\sqrt[3]{x}$	Cube root of X
Cube root	Symbol	$\sqrt[3]{}$		Cube root (8731)
cùn to m	Function (linked ; monadic)	cùn→m		Convert cùn to meter (m)
Current number of digits	Variable (longint)	#DEC	#DEC	Current number of digits as set in DISP menu (system long integer variable, write protected)
Cursor	Character	=		Character = (9255)
Cursor left	Arrow	←		Move cursor left
Cursor right	Arrow	→		Move cursor right
Cursor to begin	Command	↑ (HOME←)		Jump to top left of alpha input
Cursor to end	Command	↓ (END→)		Jump to bottom right of alpha input
cwt to kg	Function (linked ; monadic)	cwt→kg		Convert hundredweight to kilogram (kg:)
Cyclic	Character	♻		Character ♻ (9850)
D	Character	D		Character D (68)
d apostrophe lowercase	Character	d'		Character d' (271)
D caron	Character	Ď		Character Ď (270)
d lowercase	Character	d		Character d (100)
D stroke	Character	Đ		Character Đ (272)
d stroke lowercase	Character	đ		Character đ (273)
D47 layout	Layout (SIM)	D47		D47 keyboard layout ; this layout works in USER mode on the simulator (D47: Exp 2 shifts R (43S mould) /x→ R)
Date	Command	DATE	DATE	Current date (Weekday)
Date to julian day number	Function (monadic)	D→J	D→J	Convert date to julian day number (JDN) (.0 equals noon!)
Date to stack	Function (monadic)	DATE→	DATE→	Convert date to day, month, year in stack according to DISP or CLK settings for date format
Date variables	MENU	DATES	DATES	Auto-generated catalog of variables of the specified type : date
Date-time to julian day number	Function (dyadic)	DT→J	DT→J	Convert date, time in stack to julian day number (JDN) (Date, time can be in Y, X or X, Y)
Day	Command	DAY	DAY	Day (of date)
Day month year	Setting	DMY	DMY ( )	Date display mode DD.MM.YYYY (DD.MM.YYYY)
Day month year	Flag (system)	DMY		Date display mode DD.MM.YYYY (DD.MM.YYYY)
Days per year (Gregorian)	Constant (#00)	a <sub>g</sub>		yr.gregor a <sub>g</sub> = +3.652425 × 10 <sup>2</sup> (d)
dB to fld	Function (linked ; monadic)	dB→fld		Convert decibel to field ratio (fld:)

FullName	Type	Label	Catalog	Description
dB to pwr	Function (linked ; monadic)	dB→pwr		Convert decibel to power ratio (pwr:)
DBLR	Function (dyadic)	DBLR	DBLR	Double word length remainder
Decimal	Setting (pgm)	DEC	DEC	Convert X to decimal and toggle decimal mode (#TAM [LOG] (D))
Decimal	Function (monadic)	.d		Convert to decimal (real) value ; clear fraction mode, base mode ; convert degrees / hours / date to real ; convert NIM input to date (according to date format set) ; convert complex number with zero imaginary part to real number ; in PEM →REAL is entered (decimal° : ; decimal h : ; yyyy-mm-dd : )
Decimal	Function (legacy)	→REAL	→REAL	Convert to real number (Superseded by .d (keyboard g[LOG]))
Decimal	Setting (pgm)	DEC	DEC	Convert X to decimal and toggle decimal mode (#TAM [ENTER] (ENTER))
Decompose	Function (monadic)	DECOMP	DECOMP	Converts (improper) fraction to nominator in Y, denominator in X (Honours settings DENANY, DENFIX and DMX)
Decompose 3x1 matrix	Function (monadic)	M→zyx	M→zyx	Decompose 3x1 matrix to ZYX (zyx→M)
Decrement	Command	DECR	DECR	Decrement by 1 (DECR __ Tam menu)
Decrement column index	Command	J-	J-	Decrement column index J of indexed matrix ; wraps automatically ([I <sub>r</sub> J <sub>c</sub> ] = [<row> <col>])
Decrement row index	Command	I-	I-	Decrement row index I of indexed matrix ; wraps automatically ([I <sub>r</sub> J <sub>c</sub> ] = [<row> <col>])
Decrement skip on equal	Command	DSE	DSE	Decrement skip on equal (DSE __ Tam menu)
Decrement skip on less	Command	DSL	DSL	Decrement skip on less (DSL __ Tam menu)
Decrement skip on zero	Command	DSZ	DSZ	Decrement skip on zero (DSZ __ Tam menu)
Default regional formatting	Setting (pgm)	DFLT	SETDFLT	Set to default regional formats (date, time, calendar, number formatting) (First Gregorian day set: 1752-09-14)
Define menu variable	Command (PEM)	MVAR	MVAR	Define menu variable in RPN program, for VarMNU, integrator, or solver (MVAR __ Tam menu)
deg to grad	Function (linked ; monadic)	deg→grad		Convert degree to gradian (untagged) (grad:)
deg to rad	Function (linked ; monadic)	deg→rad		Convert degree to radian (untagged) (rad:)
deg/s to RPM	Function (linked ; monadic)	deg/s→RPM		Convert degree per second to rotation per minute (RPM)
Degree sign	Character	°		Character ° (176)
Degrees-radians-gradians	Function (cyclic ; monadic)	DRG	DRG	Add ADM tag to untagged value in X, convert tagged value to degrees-radians-gradians (cyclic) (Setting tag for complex X also sets POLAR mode for X according to ADM)
Delete	Command	DELETE	EQ.DEL	Delete equation
Delete	MENU	DELETE	DELETE	Commands for clearing and deleting (user) menus, variables, programs and backup files (User menus can be defined using ASSIGN)
Delete (current) program	Command	DELP	DELP	Delete current program and remove all assignments of the global labels in the program (careful!) (DELP _ TamLabel menu)
Delete all	Command	DELall	DELall	Delete all user created programs, user menus, user variables and their assignments ; clear all keyboard user assignments ; reset user configurable menus to defaults (Delete all?)
Delete all menus	Command	DELMall	DELMall	Delete all user defined menus and their assignments (Delete all user menus?)
Delete all programs	Command	DELPall	DELPall	Delete all user created programs and remove all assignments of all global labels used in the programs (Delete all programs?)
Delete all variables	Command	DELvall	DELvall	Delete all user variables and their assignments ; clear matrices HIST0 ; STATS ; Mat_A ; Mat_B ; Mat_X (Delete all user variables? ; TI : All user variables deleted)
Delete Backup	Command	DELBkup	DELBkup	Delete configuration backup file (LOAD ; SAVE) (Delete backup file?)
Delete column	Command	DELC	M.DELC	Delete column from matrix, at the cursor position
Delete items	MENU (item)	DELITM	DELITM	Delete user defined items (and their assignments), selected from category (programs - will also remove all assignments of the global labels in the program ; variables ; user defined menus) (DELITM_ menu DELITM ; Info : User menus can be defined using ASSIGN)
Delete items	MENU	DELITM		Delete user defined items (and their assignments), selected from category (programs - will also remove all assignments of the global labels in the program ; variables ; user defined menus) (User menus can be defined using ASSIGN)
Delete row	Command	DELR	M.DELR	Delete row from matrix, at the cursor position
DELTA	Character	Δ		Character Δ (alpha g[RCL] ; Code : 916)

FullName	Type	Label	Catalog	Description
delta lowercase	Character	$\delta$		Character $\delta$ (948)
Delta percent	Function (dyadic)	$\Delta\%$	$\Delta\%$	Delta percentage from Y to X, keeping Y on stack ( $\Delta\%$ :)
Delta percentage to mean	Function (monadic)	$\Delta\bar{x}$	$\Delta\bar{x}$	Delta percentage from $\bar{x}$ to x using statistics matrix (STATS) ( $\Delta\%$ :)
Delta to Star (Wye)	Function (triadic)	$\Delta \rightarrow Y$	$\Delta \rightarrow Y$	Convert star connected impedances X, Y, Z to delta impedances X, Y, Z ( $Y \rightarrow \Delta$ )
Denominator any	Setting	DENANY	DENANY [ ]	Any denominator up to D.MAX (set by DMX) may appear in fraction mode (SBI depends on SBfrac ; DENANY and DENFIX are linked (both cannot be active))
Denominator any	Flag	DENANY		Set if any denominator up to D.MAX (set by DMX) may appear in fraction mode (SBI depends on SBfrac ; DENANY and DENFIX are linked (both cannot be active))
Denominator fixed	Setting	DENFIX	DENFIX [ ]	The one and only denominator allowed in fraction mode is D.MAX (set by DMX) (SBI depends on SBfrac ; DENFIX and DENANY are linked (both cannot be active))
Denominator fixed	Flag	DENFIX		Set if the one and only denominator allowed in fraction mode is D.MAX (set by DMX) (SBI depends on SBfrac ; DENFIX and DENANY are linked (both cannot be active))
<del>Determinant</del>	<del>Function (legacy)</del>	<del>DET</del>	<del>DET</del>	<del>Determinant of matrix in X (Superseded by  M )</del>
Determinant	Function (monadic)	M	M	Determinant of matrix in X
DIGAMMA	Character	F		Character F (988)
digamma lowercase	Character	f		Character f (989)
Digit 0	Digit	0		Digit 0 (48)
Digit 1	Digit	1		Digit 1 (49)
Digit 2	Digit	2		Digit 2 (50)
Digit 3	Digit	3		Digit 3 (51)
Digit 4	Digit	4		Digit 4 (52)
Digit 5	Digit	5		Digit 5 (53)
Digit 6	Digit	6		Digit 6 (54)
Digit 7	Digit	7		Digit 7 (55)
Digit 8	Digit	8		Digit 8 (56)
Digit 9	Digit	9		Digit 9 (57)
Dimension	Function (monadic)	DIM	M.DIM	Set dimension of matrix (named variable or register) to row size (Y) and column size (X) ; creates a new matrix if necessary (M.DIM __ Tam menu)
Direct current	Character	=		Character = (9107)
Direct current	Character	=		Character = (9107)
Disk state?	Command (HW)	DISK?	DISK?	Get status of the FAT disk ; hardware only
Display (program) menu	Command (PEM)	MENU	MENU	Display the programmable menu
Display (setting for) short integer	Setting	DISP_SI		Set number of lines used for the fixed X display in BASE mode (DISP_SI _ TamNonReg menu)
Display pixel	Command	PIXEL	PIXEL	Display one pixel (X, Y)
Display point	Command	POINT	POINT	Display 9 (3x3) pixels (X, Y)
Display precision	Setting (pgm)	DSP	DSP <sub>2</sub>	Set display mode precision (only) (DSP __ TamNonReg menu)
Display settings	MENU	DISP	DISP	Display settings
Display stack registers	Setting	DSTACK	DSTACK <sub>4</sub>	Set the number of stack registers to be displayed (DSTACK _ TamNonReg menu)
Distance conversion	MENU	Dist:	Dist:	Convert between units of distance
Distributions	MENU	DISTR	DISTR	Probability distribution functions (For the probability distribution submenus the TI displays <register> = <parameter> = <value> in multi-line split screen)
Divide	Function (dyadic)	$\div$	$\div$	Divide Y by X (247)
Divide into	Function (monadic)	STO/	STO/	Divide register or variable by X (STO/ __ TamStoRcl menu)
Divides	Character			Character   (8739)
DM42 layout	Layout	DM42		DM42 keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS-R) (DM42: Final compatibility layout)
Does not contain as member	Character	∉		Character ∉ (8716)
Does not divide	Character	∤		Character ∤ (8740)
Dollar	Character	\$		Character \$ (36)

FullName	Type	Label	Catalog	Description
Double divide	Function (triadic)	DBL/	DBL/	Double word length divide (Z - least, Y - most significant digits) / (X)
Double high-reversed-9 quotation mark	Character	”		Character ” (8223)
Double integral	Character	∫		Character ∫ (8748)
Double low-9 quotation mark	Character	„		Character „ (8222)
Double multiply	Function (dyadic)	DBL×	DBL×	Double word length multiply (result in Y - least and X - most significant digit)
Down	Symbol	↓		Move down (navigation) or arrow character (alpha selection menus) (8595)
Down	Command	▼		Scroll Down Menu (or SHOW) or Single Step
Down	Arrow	↓		Move down (navigation) or arrow character (alpha selection menus)
Downwards dashed arrow	Character	↴		Character ↴ (8675)
Downwards dashed arrow	Character	↵		Character ↵ (8675)
DP decimal separator	Character	;		Character ; (9257)
DP decimal separator	Character	;		Character ; (9256)
Draw (stack limits)	MENU (item)	Draw <sub>x</sub>	Draw <sub>x</sub>	Draw graph for expression entered in NEW [EQN]
Draw (variable limits)	MENU (item)	Draw	Draw	Draw graph for expression entered in NEW [EQN]
Draw line	Setting	LINE		Connect graph points using line segments
Drop	Command	DROPx	DROPx	Drop one stack level (double [↔])
Drop Y	Command	DROPy	DROPy	Drop Y from stack
E	Character	E		Character E (69)
E acute	Character	É		Character É (201)
e acute lowercase	Character	é		Character é (233)
E breve	Character	Ĕ		Character Ĕ (276)
e breve lowercase	Character	ĕ		Character ĕ (277)
E caron	Character	Ě		Character Ě (282)
e caron lowercase	Character	ě		Character ě (283)
E circumflex	Character	Ê		Character Ê (202)
e circumflex lowercase	Character	ê		Character ê (234)
E diaeresis	Character	Ë		Character Ë (203)
e diaeresis lowercase	Character	ë		Character ë (235)
E dot	Character	Ď		Character Ď (278)
e dot lowercase	Character	ď		Character ď (279)
E grave	Character	È		Character È (200)
e grave lowercase	Character	è		Character è (232)
e lowercase	Character	e		Character e (101)
E macron	Character	Ē		Character Ē (274)
e macron lowercase	Character	ē		Character ē (275)
E ogonek	Character	Ę		Character Ę (280)
e ogonek lowercase	Character	ę		Character ę (281)
e to the power x	Function (monadic)	e <sup>x</sup>	e <sup>x</sup>	Raise e to the power in the X-register
e to the power x	Symbol	e <sup>x</sup>		e to the power x
E47 layout	Layout (SIM)	E47		E47 keyboard layout ; this layout works in USER mode on the simulator (E47: Exp 2 shifts L /x-+ R)
e <sup>x</sup> - 1	Function (monadic)	e <sup>x</sup> -1	e <sup>x</sup> -1	More accurate calculation of e <sup>x</sup> - 1 for x ≈ 0
Earth orbit (semi major axis)	Constant (#03)	a <sub>⊕</sub>		orb.earth a <sub>⊕</sub> = +1.495979 × 10 <sup>11</sup> (m)
ECA	User Program	ECA		ECA test program, preloaded from testPgms.bin
Edit equation	Command (submnu)	EDIT	EQ.EDI	Edit equation (previous equation loaded) (EIM = Equation Input Mode ; Ref : Equation editor)
Edit equation	MENU	EDIT		Edit equation (Constant names cannot be used as variables ; mode : EIM = Equation Input Mode ; starts lowercase ; [EEEX] enters E as a shortcut for 10 <sup>^</sup> ; Ref : Equation editor)
Edit matrix	MENU (item)	EDIT	M.EDI	Edit matrix in X (CAT.MENUS M.EDIT)
Edit matrix (named)	Command (submnu)	EDITN	M.EDIN	Edit matrix in a variable or register (M.EDIN __ Tam menu)
Effective annual rate	Function (niladic)	EFF/a		Calculate effective annual rate (EFF%/a = EFF%YR = EAR = )

FullName	Type	Label	Catalog	Description
Eigenvalue	Function (monadic)	EIGVAL	EIGVAL	Eigenvalue of matrix in X
Eigenvector	Function (monadic)	EIGVEC	EIGVEC	Eigenvector of matrix in X
Electrical engineering	MENU (item)	ELEC		Electrical engineering functions and custom programs (CAT.MENUS ELEC)
Electrical engineering	MENU	ELEC	ELEC	Electrical engineering functions and custom programs
Electron magnetic moment	Constant (#66)	$\mu_e$		mgmom.elec $\mu_e = -9.2847647043 \times 10^{-24}$ (J/T)
Electron magnetic moment / Bohr's magneton	Constant (#67)	$\mu_e/\mu_B$		r.elec.bohr $\mu_e/\mu_B = -1.00115965218128$
Electron rest mass	Constant (#23)	$m_e$		mass.elec $m_e = +9.1093837015 \times 10^{-31}$ (kg)
Element of	Character	$\epsilon$		Character $\epsilon$ (8712)
Elementary charge	Constant (#07)	e		charge.elem e = +1.602176634 $\times 10^{-19}$ (As)
Ellipsis	Character	...		Character ... (8230)
Elliptic amplitude	Function (dyadic)	$\psi(u,m)$	$\psi(u,m)$	Elliptic amplitude (u = X ; m = Y)
Elliptic cosine	Function (dyadic)	cn(u,m)	cn(u,m)	Elliptic cosine (u = X ; m = Y)
Elliptic delta amplitudinis	Function (dyadic)	dn(u,m)	dn(u,m)	Elliptic delta amplitudinis (u = X ; m = Y)
Elliptic sine	Function (dyadic)	sn(u,m)	sn(u,m)	Elliptic sine (u = X ; m = Y)
Elliptical	MENU	Ellipt	ELLIPT	Elliptical functions
Empty set	Character	$\emptyset$		Character $\emptyset$ (8709)
End	Command (PEM)	END	END	End statement
Energy conversion	MENU	Energy:	Energy:	Convert between units of energy
Energy equivalent of $m_u$	Constant (#31)	$m_u c^2$		energy.atom $m_u c^2 = +1.4924180856 \times 10^{-10}$ (J)
Engineering display large reals	Setting (pgm)	ENGOVR	ENGOVR ( )	Change display to ENG for reals too large to display in full
Engineering notation	Setting (pgm)	ENG	ENG ( )	Set numeric display mode to ENGINEERING notation with nn+1 digits (ENG __ TamNonReg menu)
Enter	Command	ENTER	ENTER↑	Enter input value to X (optionally also to Y) or push/duplicate value already in X to Y
Enter exponent	Command	EEX		Enter EXponent (decimal input, powers of 10) (Equation editor : [EEX] enters E as a shortcut for 10^)
Entry RPN mode	Setting	eRPN	eRPN [.]	Set Entry RPN mode
Entry RPN mode?	Command	eRPN?	eRPN?	Get parameter for Entry RPN mode ; returns 1 for ON (Set by eRPN ; to be used in programs)
Entry RPN off	Setting (pgm)	eRPNoff	RPN	Set stack to classic RPN mode (For programming purposes)
Entry RPN on	Setting (pgm)	eRPNon	eRPN	Set stack to entry RPN mode (no DUP) (For programming purposes)
EPSILON	Character	E		Character E (917)
epsilon lowercase	Character	$\epsilon$		Character $\epsilon$ (949)
epsilon tonos lowercase	Character	$\acute{\epsilon}$		Character $\acute{\epsilon}$ (941)
Equal	Symbol	=		Equal (alpha f[+] ; Code : 61)
Equation	MENU	EQN	EQN	Equation editor (Constant names cannot be used as variables ; mode : EIM = Equation Input Mode ; starts lowercase ; [EEX] enters E as a shortcut for 10^)
Error	Command (PEM)	ERR	ERR	Raise error and show error message (ERR __ TamNonReg menu)
Error function	Function (monadic)	erf	erf	Error function
Estimates	Character	$\hat{=}$		Character $\hat{=}$ (8793)
ETA	Character	H		Character H (919)
eta lowercase	Character	$\eta$		Character $\eta$ (951)
eta tonos lowercase	Character	$\acute{\eta}$		Character $\acute{\eta}$ (942)
ETH	Character	$\eth$		Character $\eth$ (208)
eth lowercase	Character	$\eth$		Character $\eth$ (240)
Euclidean norm	Function (monadic)	ENORM	ENORM	Euclidean norm of matrix in X
Euler-Mascheroni constant	Constant (#57)	$\gamma_{EM}$		c.eul.masc $\gamma_{EM} = +5.772156649015328606065120900824024 \times 10^{-1}$
Euler's Beta function	Function (dyadic)	$\beta(x,y)$	$\beta(x,y)$	Euler's Beta function
Euler's e	Constant (#08)	e		e.euler e = +2.718281828459045235360287471352662
Euler's e	Character	e		Character e (alpha f[-] ; Code : 8519)
Euler's formula	Function (monadic)	$e^{ix}$	$e^{ix}$	Rotate complex unit vector by X radians : $e^{ix} = \cos(x) + i \sin(x)$ (i or j displayed in stack and on soft buttons)
Euro	Character	€		Character € (8364)

FullName	Type	Label	Catalog	Description
European formatting	Setting (pgm)	EUROPE	SETEUR	Set to European regional formats (date, time, calendar, number formatting) (First Gregorian day set: 15.10.1582)
Even?	Function (monadic)	EVEN?	EVEN?	Test X is integer AND even
Exchange real and imaginary part	Command	Re $\leftrightarrow$ Im	Re $\leftrightarrow$ Im	Exchange real and imaginary part
Exclamation mark	Symbol	!		In edit equation, used as Factorial x! ; $\Gamma(x+1)$ (alpha f [R/S] ; Code : 33)
Exclusive NOR	Function (dyadic)	XNOR	XNOR	Logical exclusive NOR (bitwise)
Execute	Command	XEQ	XEQ	Execute function or program (XEQ __ TamLabel menu)
Execute and skip	Command (PEM)	XEQ.SKP	XEQ.SKP	Execute a (sub)routine and skip the instruction following the calling XEQ.SKP (intended use is to allow a test in the program to execute one or the other routine without the need for GTO)
Execute XEQM command	Command	X.XEQ	X.XEQ	Execute XEQM command in X-register
Exit	Command	EXIT		EXIT
Exit all	Command	EXITall	EXITALL	Exit all menus ; return to MyMenu (interactive mode) ; exit VarMNU (program mode)
Expanded inverse tangent	Function (dyadic)	ATAN2	ATAN2	Arc tangent of Y / X
Exponent	Symbol	^		Raise to power (alpha g [1/x] ; Code : 94)
Exponent	Function (monadic)	EXPT	EXPT	Exponent of number in the X-register
Exponent mark	Character	10		Character 10 (9341)
Exponent sign (AIM)	Character	<E>		Character E (displays as outline E in numeric font) (alpha g [EEX] ; Code : 8307)
Exponential	MENU	EXP	EXP	Exponential functions
Exponential (quantile)	Function (monadic)	Expon <sup>-1</sup>	Expon <sup>-1</sup>	Exponential inverse cumulative distribution
Exponential cdf (lower)	Function (monadic)	Expon <sub>l</sub>	Expon <sub>l</sub>	Exponential cumulative distribution (lower tail)
Exponential cdf (upper)	Function (monadic)	Expon <sub>u</sub>	Expon <sub>u</sub>	Exponential cumulative distribution (upper tail)
Exponential distribution	MENU	Expon:	Expon:	Exponential probability distribution (continuous) (RegR = $\lambda$ = rate parameter)
Exponential fit	Setting	ExpF	ExpF [ ]	Set BestF to include exponential curve fitting ; deselects OrthoF (Adds 2 to BestF ; TI : L.R. selected to <BestF>)
Exponential pdf	Function (monadic)	Expon <sub>p</sub>	Expon <sub>p</sub>	Exponential probability density function
Export program	Command	XPORTP	XPORTP	Export program to text file in FAT (XPORTP _ TamLabel ; DMCP : File save dialog (PROGRAMS/) ; TI : Saved)
Extended functions	MENU	X.FN	X.FN	Extended functions (Bessel, Bernoulli, Gamma, Elliptical, Orthogonal, etc.)
F	Character	F		Character F (70)
f lowercase	Character	f		Character f (102)
f''(x)	Function (monadic)	f''(x)	f''(x)	Value of 2nd derivative at x (f''(x) __ TamLabel menu ; TI : f'' =)
f'(x)	Function (monadic)	f'(x)	f'(x)	Value of 1st derivative at x (f'(x) __ TamLabel menu ; TI : f' =)
Fact	User Program	Fact		Fact test program, preloaded from testPgms.bin
Factorial x! ; $\Gamma(x+1)$	Function (monadic)	x!	x!	For integers : x! ; for reals : $\Gamma(x+1)$ (Max integer: 450 ; max real : 2123.549 956 662 463 236 31 ; integers > max are converted to reals)
Faraday constant	Constant (#09)	F		c.faraday F = +9.64853321233100184 $\times 10^4$ (As/mol)
fathom to m	Function (linked ; monadic)	fathom $\rightarrow$ m		Convert fathom to meter (m)
Feigenbaum alpha	Constant (#10)	F <sub><math>\alpha</math></sub>		$\alpha$ .feigenbm F <sub><math>\alpha</math></sub> = +2.502907875095892822283902873218216
Feigenbaum delta	Constant (#11)	F <sub><math>\delta</math></sub>		$\delta$ .feigenbm F <sub><math>\delta</math></sub> = +4.669201609102990671853203820466202
femto	Command (nonpgm)	-f		Factor $10^{-15}$
fēn to m	Function (linked ; monadic)	fēn $\rightarrow$ m		Convert fēn to meter (m)
Fibonacci	Function (monadic)	FIB	FIB	Fibonacci number n, where n = X
Fill stack	Command	FILL	FILL	Fill stack with value in the X-register
Financial	MENU	FIN	FIN	Financial calculations including time value of money (TVM)
Fine-structure constant	Constant (#55)	$\alpha_F$		c.finestruc $\alpha_F$ = +7.2973525693 $\times 10^{-3}$
fir to kg	Function (linked ; monadic)	fir $\rightarrow$ kg		Convert firkin to kilogram (kg)
fir to lb.	Function (linked ; monadic)	fir $\rightarrow$ lb.		Convert firkin to pound (lb.)
First derivative	MENU	f'	f'	First derivative menu

FullName	Type	Label	Catalog	Description
Fisher's F (quantile)	Function (monadic)	$F^{-1}(p)$	$F^{-1}(p)$	Fisher's F inverse cumulative distribution
Fisher's F cdf (lower)	Function (monadic)	$F_{\Delta}(x)$	$F_{\Delta}(x)$	Fisher's F cumulative distribution (lower tail)
Fisher's F cdf (upper)	Function (monadic)	$F_{\Delta}(x)$	$F_{\Delta}(x)$	Fisher's F cumulative distribution (upper tail)
Fisher's F distribution	MENU	F:	F:	Fisher's F probability distribution (continuous) (RegM = $d_1$ = degree of freedom ; RegN = $d_2$ = degree of freedom)
Fisher's F pdf	Function (monadic)	$F_p(x)$	$F_p(x)$	Fisher's F probability density function
Fixed notation	Setting (pgm)	FIX	FIX ( )	Set numeric display mode to FIXed notation with nn+1 digits (FIX __ TamNonReg menu)
Fixed-Scientific-Engineering	Setting (cyclic)	FSE	FSE <sub>ALL</sub>	Display mode cycling (FIX, SCI, ENG, ...) (FSE <sub>FIX</sub> ; FSE <sub>SET</sub> ; FSE <sub>ENG</sub> ; FSE <sub>UNIT</sub> ; FSE <sub>SIG</sub> ; FSE <sub>ALL</sub> )
Flag A	Shortcut (TAM)	A		Easy access to flag A (221) : Flag A (FF {A} (TAM [Σ+]))
Flag B	Shortcut (TAM)	B		Easy access to flag B (222) : Flag B (FF {B} (TAM [1/x]))
Flag browser	Browser	FLGS	FLGS	Show all flags on one page (0 = clear, 1 = set) ; show status page(s) on Up/Dn (Compare FLAGS.STATUS)
Flag C	Shortcut (TAM)	C		Easy access to flag C (223) : Flag C (FF {C} (TAM [√x]))
Flag clear?	Command	FC?	FC?	Test flag clear? (FC? __ TamFlag menu)
Flag clear? and clear	Command	FC?C	FC?C	Test flag clear? and clear (FC?C __ TamFlag menu)
Flag clear? and flip	Command	FC?F	FC?F	Test flag clear? and flip (FC?F __ TamFlag menu)
Flag clear? and set	Command	FC?S	FC?S	Test flag clear? and set (FC?S __ TamFlag menu)
Flag D	Shortcut (TAM)	D		Easy access to flag D (224) : Flag D (FF {D} (TAM [LOG]))
Flag E	Shortcut (TAM)	E		Easy access to flag E (200) : Flag E (FF {E} (TAM [LN]))
Flag F	Shortcut (TAM)	F		Easy access to flag F (201) : Flag F (FF F ([FLG.2] F2))
Flag G	Shortcut (TAM)	G		Easy access to flag G (202) : Flag G (FF {G} (TAM [STO]))
Flag H	Shortcut (TAM)	H		Easy access to flag H (203) : Flag H (FF {H} (TAM [RCL]))
Flag I	Shortcut (TAM)	I		Easy access to flag I (204) : Flag I (FF {I} (TAM [R+]))
Flag J	Shortcut (TAM)	J		Easy access to flag J (205) : Flag J (FF {J} (TAM [SIN]))
Flag K	Shortcut (TAM)	K		Easy access to flag K (206) : Flag K (FF {K} (TAM [COS]))
Flag L	Shortcut (TAM)	L		Easy access to flag L (225) : Flag L (FF {L} (TAM [TAN]))
Flag M	Shortcut (TAM)	M		Easy access to flag M (207) : Flag M (FF {M} (TAM [xzy]))
Flag N	Shortcut (TAM)	N		Easy access to flag N (208) : Flag N (FF {N} (TAM [CHS]))
Flag O	Shortcut (TAM)	O		Easy access to flag O (209) : Flag O (FF {O} (TAM [EEX]))
Flag P	Shortcut (TAM)	P		Easy access to flag P (210) : Flag P (FF P ([FLG.1] F3))
Flag Q	Shortcut (TAM)	Q		Easy access to flag Q (211) : Flag Q (FF Q ([FLG.1] F4))
Flag R	Shortcut (TAM)	R		Easy access to flag R (212) : Flag R (FF R ([FLG.1] F5))
Flag S	Shortcut (TAM)	S		Easy access to flag S (213) : Flag S (FF S ([FLG.1] F6))
Flag set?	Command	FS?	FS?	Test flag set? (FS? __ TamFlag menu)
Flag set? and clear	Command	FS?C	FS?C	Test flag set? and clear (FS?C __ TamFlag menu)
Flag set? and flip	Command	FS?F	FS?F	Test flag set? and flip (FS?F __ TamFlag menu)
Flag set? and set	Command	FS?S	FS?S	Test flag set? and set (FS?S __ TamFlag menu)
Flag T	Shortcut (TAM)	T		Easy access to flag T (220) : Flag T (FF T (TAM))
Flag U	Shortcut (TAM)	U		Easy access to flag U (214) : Flag U (FF U ([FLG.2] F6))
Flag V	Shortcut (TAM)	V		Easy access to flag V (215) : Flag V (FF V ([FLG.2] fF1))
Flag W	Shortcut (TAM)	W		Easy access to flag W (216) : Flag W (FF W ([FLG.2] fF2))
Flag X	Shortcut (TAM)	X		Easy access to flag X (217) : Flag X (FF X (TAM))
Flag Y	Shortcut (TAM)	Y		Easy access to flag Y (218) : Flag Y (FF Y (TAM))
Flag Z	Shortcut (TAM)	Z		Easy access to flag Z (219) : Flag Z (FF Z (TAM))
Flags	MENU	FLAG	FLAG	Setting, clearing and testing flags
Flattening factor	Constant (#49)	$Sf^{-1}$		f.flatteng $Sf^{-1} = +2.98257223563 \times 10^2$
fld to dB	Function (linked ; monadic)	fld→dB		Convert field ratio to decibel (dB:)
Flip bit	Function (monadic)	FB	FB	Flip bit n (FB __ TamNonReg menu)
Flip case (one character)	Alpha-shift	<f>lipchar		Flip case (one character) (alpha f + <char>)
Flip flag	Command	FF	FF	Flip flag (toggle) (FF __ TamFlag menu)

FullName	Type	Label	Catalog	Description
Floor	Function (monadic)	floor	floor	Floor (type real)
floz <sub>UK</sub> to ml	Function (linked ; monadic)	floz <sub>UK</sub> →ml		Convert UK fluid ounce to milliliter (ml:)
floz <sub>US</sub> to ml	Function (linked ; monadic)	floz <sub>US</sub> →ml		Convert US fluid ounce to milliliter (ml:)
Font browser	Browser	FBR	FBR	Browse system fonts (character tables)
For all	Character	∀		Character ∀ (8704)
Force & Pressure conversion	MENU	F&p:	F&p:	Convert between units of force and pressure
Force autoscale	Setting	X:Y=1		PLOTMENU only : force the autoscales on x- and y-axis to be the same
FP group size	Setting (pgm)	FPGRP	FPGRP <sub>3</sub>	Set fractional part group size (2..9) (FPGRP _ TamNonReg menu)
FP separator <none>	Setting (pgm)	NONE	FNONE ( )	Set fractional part separator to <none> (Menu shows symbol ø)
FP separator comma	Setting (pgm)	COM,	FCOM, ( )	Set fractional part separator to comma
FP separator dot	Setting (pgm)	DOT·	FDOT· ( )	Set fractional part separator to dot
FP separator double space	Setting (pgm)	WSPC <sub>..</sub>	FWSPC <sub>..</sub> ( )	Set fractional part separator to double space
FP separator narrow space	Setting (pgm)	NSPC <sub>.</sub>	FNSPC <sub>.</sub> ( )	Set fractional part separator to narrow space
FP separator narrow tick	Setting (pgm)	TICK'	FTICK' ( )	Set fractional part separator to narrow tick
FP separator period	Setting (pgm)	PER.	FPER. ( )	Set fractional part separator to period
FP separator space	Setting (pgm)	SPC <sub>.</sub>	FSPC <sub>.</sub> (·)	Set fractional part separator to space
FP separator tick	Setting (pgm)	WTICK'	FWTICK' ( )	Set fractional part separator to tick
FP separator underscore	Setting (pgm)	UNDR_	FUNDR_ ( )	Set fractional part separator to underscore
FP separator wide comma	Setting (pgm)	WCOM,	FWCOM, ( )	Set fractional part separator to wide comma
FP separator wide dot	Setting (pgm)	WDOT·	FWDOT· ( )	Set fractional part separator to wide dot
FP separator wide period	Setting (pgm)	WPER.	FWPER. ( )	Set fractional part separator to wide period
Fraction (mode)	Setting (cyclic ; stack)	a <sup>b</sup> / <sub>c</sub>		Set and cycle fraction mode : proper, improper fractions (tolerance determined by DMX) ; or fractional approximations of irrationals ; exit mode via [.d] (g[LOG]) ; starts in mode last used or as set by flags PROPF and IRFRAC ; when flag FRCYC is set, full cycle is available, when clear, flag PROPF is excluded (used as is) and OFF state is included in cycle (double [.] ; Info : Prefix "<" or ">" + <fraction> ; /n or /max denotes maximum denominator (set by DMX) ; Prefix "≈" + <approximation> ; SBI depends on SBfrac ; Ref : Fractions)
Fraction 1/2	Character	½		Character ½ (189)
Fraction 1/4	Character	¼		Character ¼ (188)
Fraction mode full cycle	Flag	FRCYC		Fraction mode full cycle on key [a <sup>b</sup> / <sub>c</sub> ] (g[Σ+]) ; when flag FRCYC is set, full cycle is available, when clear, flag PROPF is excluded (used as is) and OFF state is included in cycle
Fraction show register name	Setting	FRCSRN	FRCSRN [ ]	Fractions are shown with register names (x, y, z, t < or = or > or ≈)
Fraction show register name	Flag	FRCSRN		Fractions are shown with register names (x, y, z, t < or = or > or ≈)
Fractional part	Function (monadic)	FP	FP	Fractional part (#TAM [XEQ] (F) for closed number)
Fractional part?	Function (monadic)	FP?	FP?	Test X has nonzero fractional part
Fractions	Flag	FRACT		Fraction mode : proper, improper fractions (tolerance determined by DMX) (Prefix "<" or ">" + <fraction> ; /n or /max denotes maximum denominator (set by DMX) ; SBI depends on SBfrac)
Fractions	Setting	FRACT	FRACT [ ]	Fraction mode : proper, improper fractions (tolerance determined by DMX) (Prefix "<" or ">" + <fraction> ; /n or /max denotes maximum denominator (set by DMX) ; SBI depends on SBfrac)
FreeF	User Program	FreeF		FreeF test program, preloaded from testPgms.bin
FreeFp	User Program	FreeFp		FreeFp test program, preloaded from testPgms.bin
FreeFp2	User Program	FreeFp2		FreeFp2 test program, preloaded from testPgms.bin
ft. to m	Function (linked ; monadic)	ft.→m		Convert foot to meter (m:)
ft. <sup>2</sup> to ha	Function (linked ; monadic)	ft. <sup>2</sup> →ha		Convert square foot to hectare (ha:)
ft. <sup>2</sup> to m <sup>2</sup>	Function (linked ; monadic)	ft. <sup>2</sup> →m <sup>2</sup>		Convert square foot to square meter (m <sup>2</sup> :)
ft/s to km/h	Function (linked ; monadic)	ft/s→km/h		Convert foot per second to kilometer per hour (km/h:)
ft/s to m/s	Function (linked ; monadic)	ft/s→m/s		Convert foot per second to meter per second (m/s:)
ftn to s	Function (linked ; monadic)	ftn→s		Convert fortnight to second (s:)
Full stop	Character	.		Character . (46)



FullName	Type	Label	Catalog	Description
Function keys full cycle	Setting	F.1234		Longpress control : full Function key longpress cycle (1 unshifted ; 2 f-shift ; 3 g-shift ; 4 NOP)
Function keys g-shortcut	Setting	g.2Tp	g.2Tp [.]	Allow double tapping the FN-keys for a g-function (Blocked for navigation FN-keys (arrows) in editors)
Function keys skip fg	Setting	F.14		Longpress control : skip f and g Function key longpress cycle (1 unshifted ; 4 NOP)
Function keys skip g	Setting	F.124		Longpress control : skip g Function key longpress cycle (1 unshifted ; 2 f-shift ; 4 NOP)
Functions	MENU (ASM)	FCNS	FCNS	Catalog of all calculator functions (Type characters 1-2 to search)
fur to m	Function (linked ; monadic)	fur→m		Convert furlong to meter (m)
fur/ftn to km/h	Function (linked ; monadic)	fur/ftn→km/h		Convert furlong per fortnight to kilometer per hour (km/h:)
fur/ftn to m/s	Function (linked ; monadic)	fur/ftn→m/s		Convert furlong per fortnight to meter per second (m/s:)
fur/ftn to mph	Function (linked ; monadic)	fur/ftn→mph		Convert furlong per fortnight to mile per hour (mph:)
Future value	Variable (real)	FV	FV	Future value (reserved real variable) ; displays as F in menu TVM when value is large (FV = Future Value = )
G	Character	G		Character G (71)
G breve	Character	Ĝ		Character Ĝ (286)
g breve lowercase	Character	ĝ		Character ĝ (287)
g lowercase	Character	g		Character g (103)
g to carat	Function (linked ; monadic)	g→carat		Convert gram to carat (carat:)
g to oz	Function (linked ; monadic)	g→oz		Convert gram to ounce (oz:)
g to tr.oz	Function (linked ; monadic)	g→tr.oz		Convert gram to troy ounce (tr.oz:)
gal <sub>UK</sub> to l	Function (linked ; monadic)	gal <sub>UK</sub> →l		Convert UK gallon to liter (l:)
gal <sub>US</sub> to l	Function (linked ; monadic)	gal <sub>US</sub> →l		Convert US gallon to liter (l:)
GAMMA	Character	Γ		Character Γ (915)
gamma lowercase	Character	γ		Character γ (947)
Gauss fit	Setting	GaussF	GaussF [ ]	Set BestF to include Gauss curve fitting ; deselects OrthoF (Adds 256 to BestF ; TI : L.R. selected to <BestF>)
g <sub>d</sub>	Function (monadic)	g <sub>d</sub>	g <sub>d</sub>	Gudermannian function
g <sub>d</sub> <sup>-1</sup>	Function (monadic)	g <sub>d</sub> <sup>-1</sup>	g <sub>d</sub> <sup>-1</sup>	Inverse Gudermannian function
Generalised extreme value (quantile)	Function (monadic)	GEV <sup>-1</sup>	GEV <sup>-1</sup>	Generalised extreme value inverse cumulative distribution
Generalised extreme value cdf (lower)	Function (monadic)	GEV <sub>▲</sub>	GEV <sub>▲</sub>	Generalised extreme value cumulative distribution (lower tail)
Generalised extreme value cdf (upper)	Function (monadic)	GEV <sub>▲</sub>	GEV <sub>▲</sub>	Generalised extreme value cumulative distribution (upper tail)
Generalised extreme value distribution	MENU	GEV:	GEV:	Generalised extreme value probability distribution (continuous) (RegM = μ = location ; RegS = σ = scale ; RegQ = ξ = shape)
Generalised extreme value pdf	Function (monadic)	GEV <sub>p</sub>	GEV <sub>p</sub>	Generalised extreme value probability density function
Geocentric gravitational constant	Constant (#16)	GM <sub>⊕</sub>		c.grav.geo GM <sub>⊕</sub> = +3.986004418 × 10 <sup>14</sup> (m <sup>3</sup> /s <sup>2</sup> )
Geometric (quantile)	Function (monadic)	Geom <sup>-1</sup>	Geom <sup>-1</sup>	Geometric inverse cumulative distribution
Geometric cdf (lower)	Function (monadic)	Geom <sub>▲</sub>	Geom <sub>▲</sub>	Geometric cumulative distribution (lower tail)
Geometric cdf (upper)	Function (monadic)	Geom <sub>▲</sub>	Geom <sub>▲</sub>	Geometric cumulative distribution (upper tail)
Geometric distribution	MENU	Geom:	Geom:	Geometric probability distribution (discrete) (RegP = p = probability)
Geometric means	Command	̄x <sub>G</sub>	̄x <sub>G</sub>	Geometric means (̄x <sub>G</sub> ; ȳ <sub>G</sub> = (2 stack levels))
Geometric pmf	Function (monadic)	Geom <sub>p</sub>	Geom <sub>p</sub>	Geometric probability mass function
Gerver constant	Constant (#81)	μ <sub>G</sub>		c.gerver μ <sub>G</sub> = +2.21953166887197 (Moving sofa problem)
Get annual interest rate	Command	RCLI/a	RCLI/a	Get annual interest rate (percentage) (I%/a:)
Get compounding period	Command	RCLcp/a	RCLcp/a	Get compounding period per annum (CPER/a:)
Get future value	Command	RCLFV	RCLFV	Get future value (FV:)
Get number of payments	Command	RCLn	RCLn	Get number of payments (NPPER:)
Get payment	Command	RCLPMT	RCLPMT	Get payment (PMT:)
Get payment period	Command	RCLpp/a	RCLpp/a	Get payment period per annum (PPER/a:)
Get present value	Command	RCLPV	RCLPV	Get present value (PV:)

FullName	Type	Label	Catalog	Description
Get submatrix	Command	GETM	M.GET	Get submatrix with X rows and Y columns from the indexed matrix starting at current element
Giga	Command (nonpgm)	-G		Factor $10^9$
Giga binary	Command (nonpgm)	-Gi		Factor $2^{30}$
Go to	Command	GTO	GTO	Go to local/global label or line (longest[XEQ] ; TAM : GTO __ TamLabel menu)
Go to label or step	Command	GTO.	GTO.	Go to label or step ; GTO.. moves the program pointer to the end of program memory (g[XEQ] (GTO) + ".") ; TAM : GTO. ____ GTO menu)
Go to label or step	Command	GTO.	GTO.	Go to label or step ; GTO.. moves the program pointer to the end of program memory ([XEQ] + ".")
Go to matrix column	Command	GOTO Column		Go to matrix column for GOTO (GOTO Column ____ TamNonReg menu)
Go to matrix element	Command	GOTO	M.GOTO	Go to matrix element using GOTO Row ; GOTO Column (GOTO Row ; GOTO Column)
Go to matrix row	Command	GOTO Row		Go to matrix row for GOTO (GOTO Row ____ TamNonReg menu)
Golden ratio	Constant (#73)	$\phi$		r.golden $\phi = +1.618033988749894848204586834365638$ ( $\phi = (1 + \sqrt{5}) / 2$ )
grad to deg	Function (linked ; monadic)	grad→deg		Convert gradian to degree (untagged) (deg:)
grad to rad	Function (linked ; monadic)	grad→rad		Convert gradian to radian (untagged) (rad:)
Grapher (EQN variables)	MENU	f Graph	f Graph	Grapher (variables) menu for the algebraic expression entered in NEW [EQN] ; enter value and press <var_n> button(s) to initialise variable(s), then press button for variable to be plotted using the drawing functions in menu Graphs ; if only one variable exists it will be selected and starred automatically
Graphics mode	Variable (longint)	GRAMOD	GRAMOD	Graphics display mode for AGRAPH (reserved long integer variable) (OR: 0 ; SET: 1 ; OFF: 2 ; XOR: 3)
Graphing...	MENU	Graphs	Graphs	Equation graphing functions
Greater or equal than	Character	$\geq$		Character $\geq$ (8805)
Greater than	Character	>		Character > (62)
Greatest common divisor	Function (dyadic)	GCD	GCD	Greatest common divisor of X and Y
Grow (matrix edit)	Setting	GROW	M.GROW ( )	Matrix edit in grow mode (SBI depends on SBmx)
Grow (matrix edit)	Flag	GROW		Matrix edit in growing mode (SBI depends on SBmx)
H	Character	H		Character H (72)
h lowercase	Character	h		Character h (104)
h stroke lowercase	Character	ħ		Character ħ (295)
ha to acre	Function (linked ; monadic)	ha→acre		Convert hectare to acre (acre:)
ha to acre <sub>us</sub>	Function (linked ; monadic)	ha→acre <sub>us</sub>		Convert hectare to US acre (acre <sub>us</sub> :)
ha to ft. <sup>2</sup>	Function (linked ; monadic)	ha→ft. <sup>2</sup>		Convert hectare to square foot (ft. <sup>2</sup> :)
ha to km <sup>2</sup>	Function (linked ; monadic)	ha→km <sup>2</sup>		Convert hectare to square kilometer (km <sup>2</sup> :)
ha to m <sup>2</sup>	Function (linked ; monadic)	ha→m <sup>2</sup>		Convert hectare to square meter (m <sup>2</sup> :)
Harmonic means	Command	$\bar{x}_H$	$\bar{x}_H$	Harmonic means ( $\bar{x}_H$ ; $\bar{y}_H = (2$ stack levels))
Hermite polynomials (physics)	Function (dyadic)	$H_{nP}$	$H_{nP}$	Hermite polynomials (physics) ( $x = X$ ; $n = Y$ )
Hermite polynomials (probability)	Function (dyadic)	$H_n$	$H_n$	Hermite polynomials (probability) ( $x = X$ ; $n = Y$ )
Hexadecimal	Setting (pgm)	HEX	HEX	Convert X to hexadecimal and toggle hexadecimal mode (#TAM [RCL] (H))
Hide small values	Setting (pgm)	HIDE	HIDE <sub>0</sub>	Set parameter to hide small real numbers or parts ; display '0.' for numbers with absolute values $< 10^{-n}$ ; $12 \leq n \leq 99$ ; useful e.g. in matrices ; reset by HIDE 0 (HIDE __ TamNonReg menu)
Hide small values	Setting (pgm)	>HIDE<		Old version of the setting (pgm) HIDE that used stack input (May be found in older user programs)
Hide small values?	Command	HIDE?	HIDE?	Get parameter to hide small real numbers or parts (Set by HIDE)
High bin	Command	↑BIN	↑BIN	High bin (nBINS : ; ↓BIN : ; ↑BIN : (3 stack levels))
Histogram	MENU (item)	HPLLOT	HPLLOT	Histogram plotting (CAT.MENUS HPLLOT)
Histogram	MENU	HIST	HIST	Histogram functions
Histogram	MENU	HPLLOT	HPLLOT	Histogram plotting

FullName	Type	Label	Catalog	Description
Histogram matrix	Variable (matrix)	HISTO	HISTO	Reserved matrix variable Histogram (HISTO) (Created from STATS using menu HIST ; cannot be deleted)
Histogram Normal	Command	HNORM	HNORM	Fit Gauss distribution through HISTO data
Histogram X	Command	HISTOX	HISTOX	Evaluate first column of STATS and store in HISTO (nBINS ; ; #BIN ; ; #BIN : (3 stack levels))
Histogram Y	Command	HISTOY	HISTOY	Evaluate second column of STATS and store in HISTO (nBINS ; ; #BIN ; ; #BIN : (3 stack levels))
HOME	MENU	HOME	HOME	User menu to quickly access user selected menus and functions ; all buttons are user assignable (paneled look) ; initially populated for basic scitech options ; reset using HOME-R (triple [f/g] (HOME.3 ON) ; longer[f/g] (HOME.3 ON))
HOME	MENU	HOME	HOME	User menu to quickly access user selected menus and functions ; all buttons are user assignable (paneled look) ; initially populated for basic scitech options ; reset using HOME-R (longer[f/g] (HOME.3 ON))
HOME menu fff shortcut	Setting	HOME.3	HOME.3 [•]	HOME menu activated by triple (or longer) shift ([f/g]) (HOME.3 and MyM.3 are linked (both cannot be active) ; functional in 1-shift layouts)
HOME menu reset	Command	HOME-R		Reset all HOME menu user assignments (C47 HOME menu reset to default)
HOME menu shown	Setting	HOMEb	HOMEb [ ]	Base HOME menu shown (when all menus are exited using EXIT) (HOMEb and MyMb are linked (both cannot be active))
Hourglass	Character	⌚		Character ⌚ (8987)
Hours	Function (monadic)	HOUR	HOUR	Hours (of time)
HP style base	Setting	BASE <sub>HP</sub>	BASE <sub>HP</sub> [•]	Set for the classic (HP) convention that all stack registers are changed at once when base mode is changed (shortint values only)
HP style base	Flag	BASE <sub>HP</sub>		Set for the classic (HP) convention that all stack registers are changed at once when base mode is changed (shortint values only)
hp <sub>UK</sub> to W	Function (linked ; monadic)	hp <sub>UK</sub> →W		Convert UK horsepower to Watt (W)
hp <sub>E</sub> to W	Function (linked ; monadic)	hp <sub>E</sub> →W		Convert electrical horsepower to Watt (W)
hp <sub>M</sub> to W	Function (linked ; monadic)	hp <sub>M</sub> →W		Convert metric horsepower to Watt (W)
Humorous conversions	MENU	FFF+;	FFF+;	Conversions to and from the furlong-firkin-fortnight (FFF) system (and beardseconds)
Hyperbolic cosine	Function (monadic)	cosh	cosh	Hyperbolic cosine
Hyperbolic fit	Setting	HypF	HypF [ ]	Set BestF to include hyperbolic curve fitting ; deselects OrthoF (Adds 32 to BestF ; TI : L.R. selected to <BestF>)
Hyperbolic sine	Function (monadic)	sinh	sinh	Hyperbolic sine
Hyperbolic tangent	Function (monadic)	tanh	tanh	Hyperbolic tangent
Hyperfine transition frequency of <sup>133</sup> Cs	Constant (#59)	Δν <sub>CS</sub>		frq.hypf.cs Δν <sub>CS</sub> = +9.19263177 × 10 <sup>9</sup> (Hz)
Hypergeometric (quantile)	Function (monadic)	Hyper <sup>-1</sup>	Hyper <sup>-1</sup>	Hypergeometric inverse cumulative distribution
Hypergeometric cdf (lower)	Function (monadic)	Hyper <sub>p</sub>	Hyper <sub>p</sub>	Hypergeometric cumulative distribution (lower tail)
Hypergeometric cdf (upper)	Function (monadic)	Hyper <sub>p</sub>	Hyper <sub>p</sub>	Hypergeometric cumulative distribution (upper tail)
Hypergeometric distribution	MENU	Hyper:	Hyper:	Hypergeometric probability distribution (discrete) (RegM = N = population size ; RegN = n = sample size ; RegQ = K = number of special items in the pool)
Hypergeometric pmf	Function (monadic)	Hyper <sub>p</sub>	Hyper <sub>p</sub>	Hypergeometric probability mass function
I	Character	I		Character I (73)
I acute	Character	İ		Character İ (205)
i acute lowercase	Character	ı		Character ı (237)
I breve	Character	Ï		Character Ï (300)
i breve lowercase	Character	ï		Character ï (301)
I circumflex	Character	Î		Character Î (206)
i circumflex lowercase	Character	î		Character î (238)
I diaeresis	Character	Ï		Character Ï (207)
i diaeresis lowercase	Character	ï		Character ï (239)
I dot	Character	İ		Character İ (304)

FullName	Type	Label	Catalog	Description
i dotless lowercase	Character	ı		Character ı (305)
I grave	Character	İ		Character İ (204)
i grave lowercase	Character	ì		Character ì (236)
i lowercase	Character	i		Character i (105)
I macron	Character	Ī		Character Ī (298)
i macron lowercase	Character	ī		Character ī (299)
I ogonek	Character	Į		Character Į (302)
i ogonek lowercase	Character	į		Character į (303)
IBessI	User Program	IBessI		IBessI test program, preloaded from testPgms.bin
IBessP	User Program	IBessP		IBessP test program, preloaded from testPgms.bin
Identical to	Character	≡		Character ≡ (8801)
Ignore one error	Flag	IGN1ER		Set for calculator ignoring just 1 arbitrary error ; subsequently clears IGN1ER
Imaginary i	Setting (pgm)	CPX <i>i</i>	CPX <i>i</i> (•)	Set for the letter i representing the imaginary number ( <i>i</i> or <i>j</i> displayed in stack and on soft buttons)
Imaginary j	Setting (pgm)	CPX <i>j</i>	CPX <i>j</i> ( )	Set for the letter j representing the imaginary number ( <i>i</i> or <i>j</i> displayed in stack and on soft buttons)
Imaginary part	Function (monadic)	Im	Im	Imaginary part of complex number
Imaginary unit	Symbol	<i>i</i>		Mathematical symbol <i>i</i> (Displayed according to flag CPX <i>j</i> )
in. to brds	Function (linked ; monadic)	in.→brds		Convert inch to beardsecond (brds:)
in. to mm	Function (linked ; monadic)	in.→mm		Convert inch to millimeter (mm:)
in.Hg to Pa	Function (linked ; monadic)	in.Hg→Pa		Convert inch of Mercury to Pascal (Pa:)
Incomplete elliptic integral (1st)	Function (dyadic)	F(φ,m)	F(φ,m)	Incomplete elliptic integral of the 1st kind (φ = X ; m = Y)
Incomplete elliptic integral (2nd)	Function (dyadic)	E(φ,m)	E(φ,m)	Incomplete elliptic integral of the 2nd kind (φ = X ; m = Y)
Increment	Command	INCR	INCR	Increment by 1 (INCR __ Tam menu)
Increment	Character	Δ		Character Δ (8710)
Increment column index	Command	J+	J+	Increment column index J of indexed matrix ; wraps automatically ([I, J <sub>c</sub> ] = [<row> <col>])
Increment row index	Command	I+	I+	Increment row index I of indexed matrix ; wraps automatically ([I, J <sub>c</sub> ] = [<row> <col>])
Increment skip on equal	Command	ISE	ISE	Increment skip on equal (ISE __ Tam menu)
Increment skip on greater	Command	ISG	ISG	Increment skip on greater (ISG __ Tam menu)
Increment skip on zero	Command	ISZ	ISZ	Increment skip on zero (ISZ __ Tam menu)
Index matrix	Command	INDEX	INDEX	Index matrix in a variable or register ; set row I to 1, column J to 1 ; clear flags WRPEDG and WRPEND ; index wraps automatically, setting these flags (INDEX __ Tam menu)
Indian formatting	Setting (pgm)	INDIA	SETIND	Set to Indian regional formats (date, time, calendar, number formatting) (First Gregorian day set: 14.09.1752)
Indirection	Command (TAM)	→		Presented in TAM menus for commands accessing indirect input (Opens menu TamNonRegInd)
Infinities?	Command	±∞?	±∞?	Show whether X-register contains positive or negative infinite value
Infinity	Character	∞		Character ∞ (8734)
Information	MENU	INFO	INFO	System information and some information about the value in the X-register
Input	Command (PEM)	INPUT	INPUT	Halt program execution, push current value and accept input for variable or register (INPUT __ Tam menu)
Input complex	Setting	i CPX		New number input is put on the stack as complex
Input longint	Setting	i LI		New numbers are put on the stack as long integer
Input longint, real	Setting	i LI/RL		Input of long integer and reals ; standard automatic determination of which, using the decimal radix as differentiator to select real otherwise long integer
Input real	Setting	i REAL		New number input is put on the stack as real
Input/Output	MENU	I/O	I/O	Input/output functions
Insert column	Command	INSC	M.INSC	Insert column into matrix, at the cursor position
Insert row	Command	INSR	M.INSR	Insert row into matrix, at the cursor position
Integer divide	Function (dyadic)	IDIV	IDIV	Integer divide
Integer divide and remainder	Function (dyadic)	IDIVR	IDIVR	Integer divide (X) and remainder (Y)

FullName	Type	Label	Catalog	Description
Integer part	Function (monadic)	IP	IP	Integer part (type real) (#TAM [R+] (I) for closed number)
Integer product (programmable)	Command	$i\prod_n$	$i\prod_n$	Integer product using specified program ; interrupt by keypress ; if MONIT is set, displays current product and iteration counter ( $i\prod_n$ __ TamLabel menu ; <from> ENTER <to> ENTER <step>)
Integer sign mode	Variable (longint)	ISM	ISM	Integer sign mode (reserved long integer variable) (SIGNMT: s ; UNSIGN: u ; 1COMPL: 1 ; 2COMPL: 2)
Integer sign mode?	Command	ISM?	ISM?	Get sign mode for short integers (Set by UNSIGN ; SIGNMT)
Integer sum (programmable)	Command	$i\sum_n$	$i\sum_n$	Integer sum using specified program ; interrupt by keypress ; if MONIT is set, displays current sum and iteration counter ( $i\sum_n$ __ TamLabel menu ; <from> ENTER <to> ENTER <step>)
Integer Z	Character	Z		Character Z (8484)
Integer?	Function (monadic)	INT?	INT?	Test X has zero fractional part
Integers	MENU	INTS	INTS	Short integer functions
Integrate	Symbol	$\int$		Integrate (8747)
Integrator (RPN program)	MENU	$\int f d$	$\int f d$	Integrator (variables) menu for the RPN program (with MVAR variables) selected with menu TamLabel ; enter value and press <var <sub>n</sub> > button(s) to set variables, then press button for variable to be used by the integrator in menu Tool $\int$ [ADV] ; if only one variable exists it will be selected and starred automatically ; in PEM, RPN program must be specified by PGMINT and variable with menu Tam ( $\int f d$ _ TamLabel menu)
Integrator (stack limits)	Command	$\int_x^y$	$\int_x^y$	Solve the integral for the expression entered in NEW [EQN] or the RPN program (with MVAR variables) selected in TAM ; tolerance set by ACC ; monitoring set by MONIT ; in PEM, RPN program must be specified by PGMINT ( $\int_x^y$ __ Tam menu (in PEM) ; TI : Accuracy $\approx$ ; $\int \approx$ (2 stack levels))
Integrator (variable limits)	Command	$\int$	$\int f d$	Solve the integral for the expression entered in NEW [EQN] or the RPN program (with MVAR variables) selected in TAM ; tolerance set by ACC ; monitoring set by MONIT ; in PEM, RPN program must be specified by PGMINT ( $\int f d$ __ Tam menu (in PEM) ; TI : Accuracy $\approx$ ; $\int \approx$ (2 stack levels))
Integrator running	Flag (system)	INTING		Integrator is running
Integrator tools	MENU	Tool $\int$	Tool $\int$	Integrator tools ; parameter settings (if MONIT is set, the integrator displays the accumulated integral sum, the current accuracy achieved, the level counter from 7 down to 0 and an iteration counter)
Integrator tools (RPN program)	MENU	Tool $\int$		Integrator tools ; parameter settings (if MONIT is set, the integrator displays the accumulated integral sum, the current accuracy achieved, the level counter from 7 down to 0 and an iteration counter)
Interquartile range	Command	$x_{IQR}$	$x_{IQR}$	Interquartile range for both X and Y ; this is equal to Q.3 - Q.1 ( $iqr_x$ ; $iqr_y$ = (2 stack levels))
Intersection	Character	n		Character n (8745)
IntP	User Program	IntP		IntP test program, preloaded from testPgms.bin
Inverse hyperbolic cosine	Function (monadic)	arcosh	arcosh	Inverse hyperbolic cosine
Inverse hyperbolic sine	Function (monadic)	arsinh	arsinh	Inverse hyperbolic sine
Inverse hyperbolic tangent	Function (monadic)	artanh	artanh	Inverse hyperbolic tangent
Invert matrix	Function (monadic)	$[M]^{-1}$	$[M]^{-1}$	Invert matrix in X (Matrix inversion can also be performed by executing $1/x$ )
Inverted exclamation mark	Character	!		Character ! (161)
Inverted question mark	Character	?		Character ? (191)
IOTA	Character	I		Character I (921)
IOTA dialytika	Character	İ		Character İ (938)
iota dialytika lowercase	Character	ı		Character ı (970)
iota dialytika tonos lowercase	Character	ĩ		Character ĩ (912)
iota lowercase	Character	ι		Character ι (953)

FullName	Type	Label	Catalog	Description
IP first group extension	Setting (pgm)	IPGRP1x	IPGRP1x <sub>0</sub>	Extend first group to allow one additional digit up to maximum specified by parameter (IPGRP1x _ TamNonReg menu)
IP first group size	Setting (pgm)	IPGRP1	IPGRP1 <sub>0</sub>	Set integer part first group size ; parameter 0 means follow IPGRP (IPGRP1 _ TamNonReg menu)
IP group size	Setting (pgm)	IPGRP	IPGRP <sub>3</sub>	Set integer part group size (2..9) ; not for SCI or ENG notation (IPGRP _ TamNonReg menu)
IP separator <none>	Setting (pgm)	NONE	INONE ( )	Set integer part separator to <none> (Menu shows symbol ø)
IP separator comma	Setting (pgm)	COM,	ICOM, ( )	Set integer part separator to comma
IP separator dot	Setting (pgm)	DOT·	IDOT· ( )	Set integer part separator to dot
IP separator double space	Setting (pgm)	WSPC <sub>..</sub>	IWSPC <sub>..</sub> ( )	Set integer part separator to double space
IP separator narrow space	Setting (pgm)	NSPC <sub>.</sub>	INSPC <sub>.</sub> ( )	Set integer part separator to narrow space
IP separator narrow tick	Setting (pgm)	TICK'	ITICK' ( )	Set integer part separator to narrow tick
IP separator period	Setting (pgm)	PER.	IPER. ( )	Set integer part separator to period
IP separator space	Setting (pgm)	SPC <sub>.</sub>	ISPC <sub>.</sub> (·)	Set integer part separator to space
IP separator tick	Setting (pgm)	WTICK'	IWTICK' ( )	Set integer part separator to tick
IP separator underscore	Setting (pgm)	UNDR_	IUNDR_ ( )	Set integer part separator to underscore
IP separator wide comma	Setting (pgm)	WCOM,	IWCOM, ( )	Set integer part separator to wide comma
IP separator wide dot	Setting (pgm)	WDOT·	IWDOT· ( )	Set integer part separator to wide dot
IP separator wide period	Setting (pgm)	WPER.	IWPER. ( )	Set integer part separator to wide period
Irrational fractions	Setting	IRFRAC	IRFRAC [ ]	Irrational fraction mode (proper, improper fractions) ; SBI·   indicates that approximate multiples or fractions are used, of the irrational numbers $\sqrt{2}$ , $\sqrt{3}$ , $\sqrt{5}$ , $\phi$ , $\pi$ and $e$ (with tolerance $10^{-24}$ ) (Prefix "≈" + <approximation> ; SBI depends on SBfrac)
Irrational I	Character	I		Character I (8464)
Iteration monitoring	Flag	MONIT		If set, iteration status information is shown while executing programmable sums, solvers and integrator
I <sub>xyz</sub>	Function (triadic)	I <sub>xyz</sub>	I <sub>xyz</sub>	Regularised (incomplete) Beta function
IF <sub>p</sub>	Function (dyadic)	IF <sub>p</sub>	IF <sub>p</sub>	Regularised Gamma function (P)
IF <sub>q</sub>	Function (dyadic)	IF <sub>q</sub>	IF <sub>q</sub>	Regularised Gamma function (Q)
J	Character	J		Character J (74)
j lowercase	Character	j		Character j (106)
J to Btu	Function (linked ; monadic)	J→Btu		Convert Joule to British Thermal Unit (Btu:)
J to cal	Function (linked ; monadic)	J→cal		Convert Joule to calorie (cal:)
J to Wh	Function (linked ; monadic)	J→Wh		Convert Joule to Watt-hour (Wh:)
Jacobi's Zeta	Function (dyadic)	Z(φ,m)	Z(φ,m)	Jacobi's Zeta (φ = X ; m = Y)
Japanese formatting	Setting (pgm)	JAPAN	SETJPN	Set to Japanese regional formats (date, time, calendar, number formatting) (First Gregorian day set: 1873-01-01)
jīn to kg	Function (linked ; monadic)	jīn→kg		Convert jīn to kilogram (kg:)
Josephson constant	Constant (#21)	K <sub>J</sub>		c.josephsn K <sub>J</sub> = +4.835978484169836324476582850545281 × 10 <sup>14</sup> (Hz/V)
Julian day number to date-time	Command	J→DT	J→DT	Convert julian day number (JDN) to date, time in stack
Julian-Gregorian transition 1582	Setting (pgm)	JG.1582	JG.1582	Set Julian-Gregorian transition date to 1582-10-15 (First Gregorian day set: 1582-10-15)
Julian-Gregorian transition 1752	Setting (pgm)	JG.1752	JG.1752	Set Julian-Gregorian transition date to 1752-09-14 (First Gregorian day set: 1752-09-14)
Julian-Gregorian transition 1873	Setting (pgm)	JG.1873	JG.1873	Set Julian-Gregorian transition date to 1873-01-01 (First Gregorian day set: 1873-01-01)
Julian-Gregorian transition 1949	Setting (pgm)	JG.1949	JG.1949	Set Julian-Gregorian transition date to 1949-10-01 (First Gregorian day set: 1949-10-01)
Julian-Gregorian? (transition)	Command	J/G?	J/G?	Get the day that Julian date changes over to Gregorian date (Set by J/G)
J <sub>y</sub> (x)	Function (monadic)	J <sub>y</sub> (x)	J <sub>y</sub> (x)	Bessel function of the 1st kind and order y
K	Character	K		Character K (75)
K	User Program	K		K test program, preloaded from testPgms.bin
k lowercase	Character	k		Character k (107)
KAPPA	Character	K		Character K (922)
kappa lowercase	Character	κ		Character κ (954)
Key	Command (PEM)	KEY	KEY	Used in programs to form program step for KEYG → KEY nn GT0 __ or KEYX → KEY nn XEQ __ (KEY __ TamNonReg menu)

FullName	Type	Label	Catalog	Description
Key execute	Command (PEM)	KEYX	KEYX	Used in programs to specify label to execute when soft button (1-18), [▲] (19), [▼] (20) or [EXIT] (21) is pressed (KEY __ TamKey menu ; KEY XEQ __ TamLabel menu)
Key go to	Command (PEM)	KEYG	KEYG	Used in programs to specify label to go to when soft button (1-18), [▲] (19), [▼] (20) or [EXIT] (21) is pressed (KEY __ TamKey menu ; KEY GTO __ TamLabel menu)
Key pressed?	Command	KEY?	KEY?	Test key was pressed (store keycode in register) (KEY? __ Tam menu)
Key type?	Function (monadic)	KTYP?	KTYP?	Get key type for key code returned by KEY? (KTYP? __ Tam menu)
Keyboard (USER) reset	Command	KEYS-R		Delete all keyboard user assignments (C47 USER keys cleaned)
Keyboard map	Browser	KEYMAP	KEYMAP	Show keyboard map for USER mode ; press and hold key [.] to view normal mode (STD) ; also showing special assignment from menu -> +NRM (Browse normal and alpha key shifts using [▲] and [▼])
Keys	MENU	KEYS	KEYS	Layouts, ribbons and special key assignments (Switching (hardware) layouts deletes all keyboard user assignments! (use SAVE to backup))
kg to cwt	Function (linked ; monadic)	kg→cwt		Convert kilogram to hundredweight (cwt:)
kg to fir	Function (linked ; monadic)	kg→fir		Convert kilogram to firkin (fir:)
kg to jīn	Function (linked ; monadic)	kg→jīn		Convert kilogram to jīn (jīn:)
kg to lb.	Function (linked ; monadic)	kg→lb.		Convert kilogram to pound (lb.:)
kg to liǎng	Function (linked ; monadic)	kg→liǎng		Convert kilogram to liǎng (liǎng:)
kg to short cwt	Function (linked ; monadic)	kg→short cwt		Convert kilogram to short hundredweight (short cwt:)
kg to short ton	Function (linked ; monadic)	kg→short ton		Convert kilogram to short ton (short ton:)
kg to stone	Function (linked ; monadic)	kg→stone		Convert kilogram to stone (stone:)
kg to ton	Function (linked ; monadic)	kg→ton		Convert kilogram to ton (ton:)
Khinchin constant	Constant (#56)	$K_0$		c.khinchin $K_0 = +2.685452001065306445309714835481796$
kilo	Command (nonpgm)	-k		Factor $10^3$
Kilo binary	Command (nonpgm)	-ki		Factor $2^{10}$
km to mi.	Function (linked ; monadic)	km→mi.		Convert kilometer to mile (mi.:)
km to nmi	Function (linked ; monadic)	km→nmi		Convert kilometer to nautical mile (nmi:)
km/h to ft/s	Function (linked ; monadic)	km/h→ft/s		Convert kilometer per hour to foot per second (ft/s:)
km/h to fur/ftn	Function (linked ; monadic)	km/h→fur/ftn		Convert kilometer per hour to furlong per fortnight (fur/ftn:)
km/h to knot	Function (linked ; monadic)	km/h→knot		Convert kilometer per hour to knot (knot:)
km/h to m/s	Function (linked ; monadic)	km/h→m/s		Convert kilometer per hour to meter per second (m/s:)
km/h to mph	Function (linked ; monadic)	km/h→mph		Convert kilometer per hour to mile per hour (mph:)
km/kWh to kWh/100km	Function (linked ; monadic)	km/kWh→kWh/100km		Convert kilometer per kilowatt-hour to kilowatt-hour per 100 km (kWh/100km:)
km/l to l/100km	Function (linked ; monadic)	km/l→l/100km		Convert kilometer per liter to liter per 100 km (l/100km:)
km/l <sub>e</sub> to kWh/100km	Function (linked ; monadic)	km/l <sub>e</sub> →kWh/100km		Convert kilometer per liter equivalent to kilowatt-hour per 100 km (kWh/100km:)
km <sup>2</sup> to ha	Function (linked ; monadic)	km <sup>2</sup> →ha		Convert square kilometer to hectare (ha:)
km <sup>2</sup> to mi <sup>2</sup>	Function (linked ; monadic)	km <sup>2</sup> →mi <sup>2</sup>		Convert square kilometer to square mile (mi <sup>2</sup> :)
km <sup>2</sup> to nmi <sup>2</sup>	Function (linked ; monadic)	km <sup>2</sup> →nmi <sup>2</sup>		Convert square kilometer to square nautical mile (nmi <sup>2</sup> :)
knot to km/h	Function (linked ; monadic)	knot→km/h		Convert knot to kilometer per hour (km/h:)
kWh/100km to km/kWh	Function (linked ; monadic)	kWh/100km→km/kWh		Convert kilowatt-hour per 100 km to kilometer per kilowatt-hour (km/kWh:)
kWh/100km to km/l <sub>e</sub>	Function (linked ; monadic)	kWh/100km→km/l <sub>e</sub>		Convert kilowatt-hour per 100 km to kilometer per liter equivalent (km/l <sub>e</sub> :)
kWh/100km to kWh/100mi	Function (linked ; monadic)	kWh/100km→kWh/100mi		Convert kilowatt-hour per 100 km to kilowatt-hour per 100 mile ( )
kWh/100mi to kWh/100km	Function (linked ; monadic)	kWh/100mi→kWh/100km		Convert kilowatt-hour per 100 mile to kilowatt-hour per 100 km (kWh/100km:)
kWh/100mi to mi/kWh	Function (linked ; monadic)	kWh/100mi→mi/kWh		Convert kilowatt-hour per 100 mile to mile per kilowatt-hour (mi/kWh:)
kWh/100mi to mpge <sub>UK</sub>	Function (linked ; monadic)	kWh/100mi→mpge <sub>UK</sub>		Convert kilowatt-hour per 100 mile to mpg gasoline equivalent UK (mpge <sub>UK</sub> :)
kWh/100mi to mpge <sub>US</sub>	Function (linked ; monadic)	kWh/100mi→mpge <sub>US</sub>		Convert kilowatt-hour per 100 mile to mpg gasoline equivalent US (mpge <sub>US</sub> :)
L	Character	L		Character L (76)
L acute	Character	Ł		Character Ł (313)
L apostrophe	Character	Ĺ		Character Ĺ (317)
l apostrophe lowercase	Character	ĺ		Character ĺ (318)
l lowercase	Character	l		Character l (108)

FullName	Type	Label	Catalog	Description
L stroke	Character	ł		Character ł (321)
l stroke lowercase	Character	l		Character l (322)
l to gal <sub>UK</sub>	Function (linked ; monadic)	l→gal <sub>UK</sub>		Convert liter to UK gallon (gal <sub>UK</sub> .)
l to gal <sub>US</sub>	Function (linked ; monadic)	l→gal <sub>US</sub>		Convert liter to US gallon (gal <sub>US</sub> .)
l to qt.	Function (linked ; monadic)	l→qt.		Convert liter to quart (qt.:)
l.y. to m	Function (linked ; monadic)	l.y.→m		Convert lightyear to meter (m)
l/100km to km/l	Function (linked ; monadic)	l/100km→km/l		Convert liter per 100 km to kilometer per liter (km/l:)
l/100km to mpg <sub>UK</sub>	Function (linked ; monadic)	l/100km→mpg <sub>UK</sub>		Convert liter per 100 km to mile per gallon UK (mpg <sub>UK</sub> .)
l/100km to mpg <sub>US</sub>	Function (linked ; monadic)	l/100km→mpg <sub>US</sub>		Convert liter per 100 km to mile per gallon US (mpg <sub>US</sub> .)
Label	Command (PEM)	LBL	LBL	Create local/global label (LBL __ TamLabel menu)
Label exists?	Command	LBL?	LBL?	Test label exists (LBL? __ TamLabel menu)
Laguerre generalised polynomials	Function (triadic)	L <sub>mα</sub>	L <sub>mα</sub>	Laguerre generalised polynomials (x = X ; m = Y ; α = Z)
Laguerre polynomials	Function (dyadic)	L <sub>m</sub>	L <sub>m</sub>	Laguerre polynomials (x = X ; m = Y)
LAMBDA	Character	λ		Character λ (923)
lambda lowercase	Character	λ		Character λ (955)
Landé's electron g-factor	Constant (#15)	g <sub>e</sub>		gfact.elec g <sub>e</sub> = -2.00231930436256
Large longint display	Setting	LRG_LI	LRG_LI [•]	Large longint display
Last statefile loaded?	Command	STATE?	STATE?	Get name of last full statefile or backup loaded from STATE/<state>.sav, SAVFILES/C47.sav or SAVFILES/C47auto.sav (Last full state file loaded: <date>: <file> (1 stack level))
Last Ticks?	Command	LASTT?	LASTT?	Get number of ticks for the previous operation (command, function, program) (One tick is 10 ms)
Last X	Command	LASTx	LASTx	Recall last X (register L)
Layouts	MENU	LAYOUTS	LAYOUTS	Keyboard layouts to use on the hardware and in the simulator, for testing ; selection of a sim only layout sets USER mode ; selection of a hardware layout clears USER mode ; default : C47 (Switching (hardware) layouts deletes all keyboard user assignments! (use SAVE to backup))
lb. to fir	Function (linked ; monadic)	lb.→fir		Convert pound to firkin (fir:)
lb. to kg	Function (linked ; monadic)	lb.→kg		Convert pound to kilogram (kg:)
lbf to N	Function (linked ; monadic)	lbf→N		Convert poundforce to Newton (N:)
lbf-ft to Nm	Function (linked ; monadic)	lbf-ft→Nm		Convert pound-foot to Newton-meter (Nm:)
Leading zeros	Setting	LEAD.0	LEAD.0 [ ]	Leading zeros are on (shortint bases 2, 4, 8, 16)
Leading zeros	Flag	LEAD.0		Leading zeros are on (shortint bases 2, 4, 8, 16)
Leap year?	Function (monadic)	LEAP?	LEAP?	Test date is in leap year
Least common multiple	Function (dyadic)	LCM	LCM	Least common multiple of X and Y
Left	Symbol	←		Move left (navigation) or arrow character (alpha selection menus) (8592)
Left	Arrow	←		Move left (navigation) or arrow character (alpha selection menus)
Left bit mask	Function (monadic)	MASKL	MASKL	Set left n bits to use as mask (MASKL __ TamNonReg menu)
Left curly bracket	Character	{		Character { (123)
Left double quotation mark	Character	“		Character “ (8220)
Left French quotation mark	Character	«		Character « (171)
Left justify	Function (monadic)	LJ	LJ	Left justify (within word size) ; returns shift in X and result in Y
Left parenthesis	Symbol	(		Open (left) parenthesis (40)
Left right arrows	Character	↔		Character ↔ (8596)
Left right double arrows	Character	↔		Character ↔ (8660)
Left single quotation mark	Character	'		Character ' (8216)
Left square bracket	Character	[		Character [ (91)
Left-left	Arrow	←		Move all the way to the left (navigation) or arrow character (alpha selection menus)
Leftwards dashed arrow	Character	↔		Character ↔ (8672)
Leftwards dashed arrow	Character	↔		Character ↔ (8672)
Legendre polynomials	Function (dyadic)	P <sub>n</sub>	P <sub>n</sub>	Legendre polynomials (x = X ; n = Y)



FullName	Type	Label	Catalog	Description
Len2	User Program	Len2		Len2 test program, preloaded from testPgms.bin
Len3	User Program	Len3		Len3 test program, preloaded from testPgms.bin
Len4	User Program	Len4		Len4 test program, preloaded from testPgms.bin
Len5	User Program	Len5		Len5 test program, preloaded from testPgms.bin
Len6	User Program	Len6		Len6 test program, preloaded from testPgms.bin
Len7	User Program	Len7		Len7 test program, preloaded from testPgms.bin
Len8	User Program	Len8		Len8 test program, preloaded from testPgms.bin
Less or equal than	Character	≤		Character ≤ (8804)
Less than	Character	<		Character < (60)
lǐ to m	Function (linked ; monadic)	lǐ→m		Convert lǐ to meter (m)
liǎng to kg	Function (linked ; monadic)	liǎng→kg		Convert liǎng to kilogram (kg)
Linear fit	Setting	LinF	LinF [·]	Set BestF to include linear curve fitting ; deselects OrthoF (Adds 1 to BestF ; TI : L.R. selected to <BestF>)
Linear interpolation	Function (triadic)	LINPOL	LINPOL	Linear interpolation ; also works for complex numbers (<from> ENTER <towards> ENTER <at> (fraction))
Linear Regression	Command	L.R.	L.R.	Calculate all selected regression models, and select the one with the best coefficient of correlation (r)
LISSAJ	User Program	LISSAJ		LISSAJ test program, preloaded from testPgms.bin
List	Character	≡		Character ≡ (8652)
List graph coordinates	Command	LISTXY		List the actual STATS graph coordinates (7 digit floating point)
LN(1+x)	Function (monadic)	LN(1+x)	LN(1+x)	More accurate calculation of LN(1+x) for x ≈ 0
LNβ	Function (dyadic)	LNβ	LNβ	Natural logarithm of Euler's Beta function
LNΓ	Function (monadic)	LNΓ	LNΓ	Natural logarithm of the Gamma function
Load full backup	Command (nonpgm)	LOAD	LOAD	Load full backup from SAVE file SAVFILES/C47.sav in FAT and auto-clear USER mode (Backup restored + <version info> ; Directory : SAVFILES/)
Load full state	Command (nonpgm)	LOADST	LOADST	Load full state from SAVEST file STATE/<state>.s47 in FAT ; presenting "WARNING: Current calculator state will be lost. Press [ENTER] to confirm." (State file loaded ; DMCP File open dialog (STATE/))
Load program	Command (nonpgm)	LOADP	LOADP	Load program from SAVE file SAVFILES/C47.sav in FAT (Programs and equations loaded ; Directory : SAVFILES/)
Load program	Command	X.LOAD		Load XEQC program (X.LOAD __ TamNonReg menu)
Load registers	Command (nonpgm)	LOADR	LOADR	Load registers from SAVE file SAVFILES/C47.sav in FAT (Global and local registers loaded (w/ local flags) ; Directory : SAVFILES/)
Load Sigma registers	Command (nonpgm)	LOADΣ	LOADΣ	Load Sigma registers from SAVE file SAVFILES/C47.sav in FAT (Statistical data loaded ; Directory : SAVFILES/)
Load system state	Command (nonpgm)	LOADSS	LOADSS	Load system state from SAVE file SAVFILES/C47.sav in FAT (System settings loaded ; Directory : SAVFILES/)
Load variables	Command (nonpgm)	LOADV	LOADV	Load variables from SAVE file SAVFILES/C47.sav in FAT (User variables loaded ; Directory : SAVFILES/)
Log normal (quantile)	Function (monadic)	LgNrm <sup>-1</sup>	LgNrm <sup>-1</sup>	Log normal inverse cumulative distribution
Log normal cdf (lower)	Function (monadic)	LgNrm <sub>Ⓛ</sub>	LgNrm <sub>Ⓛ</sub>	Log normal cumulative distribution (lower tail)
Log normal cdf (upper)	Function (monadic)	LgNrm <sub>Ⓡ</sub>	LgNrm <sub>Ⓡ</sub>	Log normal cumulative distribution (upper tail)
Log normal pdf	Function (monadic)	LgNrm <sub>p</sub>	LgNrm <sub>p</sub>	Log normal probability density function
Logarithmic fit	Setting	LogF	LogF [ ]	Set BestF to include logarithmic curve fitting ; deselects OrthoF (Adds 4 to BestF ; TI : L.R. selected to <BestF>)
Logistic (quantile)	Function (monadic)	Logis <sup>-1</sup>	Logis <sup>-1</sup>	Logistic inverse cumulative distribution
Logistic cdf (lower)	Function (monadic)	Logis <sub>Ⓛ</sub>	Logis <sub>Ⓛ</sub>	Logistic cumulative distribution (lower tail)
Logistic cdf (upper)	Function (monadic)	Logis <sub>Ⓡ</sub>	Logis <sub>Ⓡ</sub>	Logistic cumulative distribution (upper tail)
Logistic distribution	MENU	Logis:	Logis:	Logistic probability distribution (continuous) (RegM = μ = location ; RegS = s = scale)

FullName	Type	Label	Catalog	Description
Logistic pdf	Function (monadic)	Logis <sub>p</sub>	Logis <sub>p</sub>	Logistic probability density function
LOG <sub>x</sub> y	Function (dyadic)	LOG <sub>x</sub> y	LOG <sub>x</sub> y	Logarithm of Y for base X
Long integer	Function (monadic)	LINT	LINT	Convert to long integer (max 1000 digits)
Long keypress	Setting	KEY.LP	KEY.LP [•]	Long keypress option(s) for [<f>]; [<g>]; XEQ; DRG (Functional in 2-shift layouts)
Long keypress	Flag	KEY.LP		Long keypress option(s) for [<f>]; [<g>]; XEQ; DRG (Functional in 2-shift layouts)
Longint variables	MENU	L.INTS	L.INTS	Auto-generated catalog of variables of the specified type : longint
Looping	MENU	LOOP	LOOP	Looping (programming) functions
Low Battery	Flag (system)	LOWBAT		Low battery voltage
Low bin	Command	↓BIN	↓BIN	Low bin (nBINS : ; ↓BIN : ; ↑BIN : (3 stack levels))
Lower limit	Variable (real)	↓Lim	↓LIM	Lower limit for solvers and integrator (reserved real variable) ; displays as ↓L in menu ToolS or ToolJ when value ≠ 0 ; set interactively and by realSlv <sub>x</sub> <sup>x</sup> , cpxSlv <sub>y</sub> <sup>x</sup> or ∫ <sub>x</sub> <sup>x</sup> (↓LIM :)
Lower limit of drawing interval	Variable (real)	↓X	↓X	Lower limit of drawing interval (reserved real variable) ; set interactively and by Draw <sub>x</sub> <sup>x</sup>
Lower quantile	Command	x <sub>q1</sub>	x <sub>q1</sub>	Lower quantile for both X and Y (Q <sub>1 x</sub> ; Q <sub>1 y</sub> = (2 stack levels))
LP1a	User Program	LP1a		LP1a test program, preloaded from testPgms.bin
LU decomposition	Function (monadic)	M.LU	M.LU	Decompose matrix in X into lower (L) in Y and upper (U or R) matrix in X ; pivot matrix in Z (LU (LR) factorisation)
M	Character	M		Character M (??)
m lowercase	Character	m		Character m (109)
m to au	Function (linked ; monadic)	m→au		Convert meter to astronomical unit (au:)
m to brds	Function (linked ; monadic)	m→brds		Convert meter to beardsecond (brds:)
m to chī	Function (linked ; monadic)	m→chī		Convert meter to chī (chī:)
m to cùn	Function (linked ; monadic)	m→cùn		Convert meter to cùn (cùn:)
m to fathom	Function (linked ; monadic)	m→fathom		Convert meter to fathom (fathom:)
m to fēn	Function (linked ; monadic)	m→fēn		Convert meter to fēn (fēn:)
m to ft.	Function (linked ; monadic)	m→ft.		Convert meter to foot (ft.:)
m to fur	Function (linked ; monadic)	m→fur		Convert meter to furlong (fur:)
m to l.y.	Function (linked ; monadic)	m→l.y.		Convert meter to lightyear (l.y.:)
m to lǐ	Function (linked ; monadic)	m→lǐ		Convert meter to lǐ (lǐ:)
m to mi.	Function (linked ; monadic)	m→mi.		Convert meter to mile (mi.:)
m to nmi	Function (linked ; monadic)	m→nmi		Convert meter to nautical mile (nmi:)
m to pc	Function (linked ; monadic)	m→pc		Convert meter to parsec (pc:)
m to survey ft <sub>US</sub>	Function (linked ; monadic)	m→survey ft <sub>US</sub>		Convert meter to US survey foot (survey ft <sub>US</sub> :)
m to yd.	Function (linked ; monadic)	m→yd.		Convert meter to yard (yd.:)
m to yīn	Function (linked ; monadic)	m→yīn		Convert meter to yīn (yīn:)
m to zhàng	Function (linked ; monadic)	m→zhàng		Convert meter to zhàng (zhàng:)
M.FACT <sub>x</sub>	Function (monadic)	M.FACT <sub>x</sub>	M.FACT <sub>x</sub>	Multiply (prime) factors from a 2 row matrix in X (row 1 contains the factors, row 2 contains the powers) (E.g. use matrix resulting from FACTORS as input)
m/s to ft/s	Function (linked ; monadic)	m/s→ft/s		Convert meter per second to foot per second (ft/s:)
m/s to fur/ftn	Function (linked ; monadic)	m/s→fur/ftn		Convert meter per second to furlong per fortnight (fur/ftn:)
m/s to km/h	Function (linked ; monadic)	m/s→km/h		Convert meter per second to kilometer per hour (km/h:)
m/s to mph	Function (linked ; monadic)	m/s→mph		Convert meter per second to mile per hour (mph:)
m <sup>2</sup> to ft. <sup>2</sup>	Function (linked ; monadic)	m <sup>2</sup> →ft. <sup>2</sup>		Convert square meter to square foot (ft. <sup>2</sup> :)
m <sup>2</sup> to ha	Function (linked ; monadic)	m <sup>2</sup> →ha		Convert square meter to hectare (ha:)
m <sup>2</sup> to mǔ	Function (linked ; monadic)	m <sup>2</sup> →mǔ		Convert square meter to mǔ (mǔ:)
m <sup>3</sup> to barrel	Function (linked ; monadic)	m <sup>3</sup> →barrel		Convert cubic meter to barrel (barrel:)
Magnetic flux quantum	Constant (#74)	φ <sub>0</sub>		fluxq.magn φ <sub>0</sub> = +2.067833848461929323081115412147497 × 10 <sup>-15</sup> (Vs)
Magnitude	Function (monadic)	x	x	Magnitude (absolute value) of complex number
Mantissa	Function (monadic)	MANT	MANT	Mantissa

FullName	Type	Label	Catalog	Description
Mass conversion	MENU	Mass:	Mass:	Convert between units of mass
Mass of the Earth	Constant (#34)	$M_{\oplus}$		mass.earth $M_{\oplus} = +5.9736 \times 10^{24}$ (kg)
Mass of the Moon	Constant (#24)	$M_{\text{Moon}}$		mass.moon $M_{\text{Moon}} = +7.349 \times 10^{22}$ (kg)
Mass of the Sun	Constant (#33)	$M_{\odot}$		mass.sun $M_{\odot} = +1.9891 \times 10^{30}$ (kg)
Math keys full cycle	Setting	M.1234		Longpress control : full Math keys longpress cycle (1 unshifted ; 2 f-shift ; 3 g-shift ; 4 NOP)
Math keys skip fg	Setting	M.14		Longpress control : skip f and g Math keys longpress cycle (1 unshifted ; 4 NOP)
Math keys skip g	Setting	M.124		Longpress control : skip g Math keys longpress cycle (1 unshifted ; 2 f-shift ; 4 NOP)
Matrix	MENU	MATX	MATX	Matrix functions (A matrix is displayed in X and shown in other stack registers as [MxN Matrix] or [MxN C Matrix] for a matrix containing complex element(s) ; in edit mode, only monadic functions on matrix elements are possible, use registers to input calculation results)
Matrix A	Variable (matrix)	Mat A	Mat_A	Reserved matrix variable Matrix A (Mat_A) (Enter data using menu SIM EQ)
Matrix B	Variable (matrix)	Mat B	Mat_B	Reserved matrix variable Matrix B (Mat_B) (Enter data using menu SIM EQ)
Matrix dimension? (X)	Function (monadic)	DIM?	M.DIM?	Get dimensions of matrix in X
Matrix editor	MENU	EDIT	M.EDIT	Open matrix editor menu (MIM = Matrix Input Mode ; Ref : Matrix editor)
Matrix lower left	Character			Character   (9123)
Matrix lower right	Character			Character   (9126)
Matrix middle left	Character			Character   (9122)
Matrix middle right	Character			Character   (9125)
Matrix simultaneous equations	MENU (item)	SIM EQ	SIM_EQ	Matrix simultaneous equations functions (Mat A * Mat X = Mat B) (SIM_EQ __ TamNonReg menu ; Info : CAT.MENUS M.SIMQ)
Matrix simultaneous equations	MENU	SIM EQ	M.SIMQ	Matrix simultaneous equations functions (Mat A * Mat X = Mat B)
Matrix upper left	Character			Character   (9121)
Matrix upper right	Character			Character   (9124)
Matrix variables	MENU	MATRS	MATRS	Auto-generated catalog of variables of the specified type : matrix
Matrix X	Command	Mat X	Mat_X	Create reserved matrix variable Matrix X (Mat_X) and solve simultaneous equations
Matrix X	Variable (matrix)	Mat X	Mat_X	Reserved matrix variable Matrix X (Mat_X) (Solve equations using menu SIM EQ)
Matrix?	Function (monadic)	MATR?	MATR?	Test X is a matrix
Maxima	Command	$x_{\text{MAX}}$	$x_{\text{MAX}}$	Maxima ( $x_{\text{MAX}}$ ; $y_{\text{MAX}} = (2 \text{ stack levels})$ )
Maximum	Function (dyadic)	max	max	Maximum of X and Y
Maximum denominator	Setting (pgm)	DMX	DMX <sub>64</sub>	Set maximum denominator for fractions ; default shown as 1/64 (DMX ____ TamNonReg menu)
Maximum denominator	Variable (longint)	D.MAX	D.MAX	Maximum denominator (system long integer variable, write protected)
Maximum denominator	Setting (pgm)	>DMX<		Old version of the setting (pgm) DMX that used stack input (May be found in older user programs)
Maximum denominator?	Command	DMX?	DMX?	Get maximum denominator for fractions (Set by DMX)
Mean radius of the Earth	Constant (#44)	$R_{\oplus}$		rad.earth $R_{\oplus} = +6.37101 \times 10^6$ (m)
Mean radius of the Moon	Constant (#41)	$R_{\text{Moon}}$		rad.moon $R_{\text{Moon}} = +1.73753 \times 10^6$ (m)
Mean radius of the Sun	Constant (#43)	$R_{\odot}$		rad.sun $R_{\odot} = +6.96 \times 10^8$ (m)
Mean rate of return	Command	%MRR	%MRR	Mean rate of return in percentage per period (% :)
Measured angle	Character	∠		Character ∠ (8737)
Median	Command	$x_{\text{MEDN}}$	$x_{\text{MEDN}}$	Sort the data and return the middle value for both X and Y ; for an even number of samples, the arithmetic mean of the two middle values is returned ( $md_x$ ; $md_y = (2 \text{ stack levels})$ )
Median absolute deviation	Command	$x_{\text{MAD}}$	$x_{\text{MAD}}$	Median absolute deviation for both X and Y ; this is the median of the differences between each data point and the overall median ( $mad_x$ ; $mad_y = (2 \text{ stack levels})$ )
Mega	Command (nonpgm)	-M		Factor $10^6$
Mega binary	Command (nonpgm)	-Mi		Factor $2^{20}$
Memory? (RAM)	Command	MEM?	MEM?	Get amount of free RAM memory, in bytes (Bytes =)
Menu fg-highlighting Full	Setting	fg.FUL	fg.FUL (•)	Show full horizontal f- and g-lines indicating the state of the f- or g-shift in menu
Menu fg-highlighting Limited	Setting	fg.LIM	fg.LIM ( )	Show limited horizontal f- and g-lines indicating the state of the f- or g-shift in menu
Menu fg-highlighting Off	Setting	fg.OFF	fg.OFF ( )	Do not show horizontal f- and g-lines indicating the state of the f- or g-shift in menu

FullName	Type	Label	Catalog	Description
MENUS	MENU (ASM)	MENUS		Catalog of all menus (including user defined menus) (Type characters 1-2 to search)
Message	Command	MSG	MSG	Show error message (MSG __ Tam menu)
mi. to km	Function (linked ; monadic)	mi.→km		Convert mile to kilometer (km)
mi. to m	Function (linked ; monadic)	mi.→m		Convert mile to meter (m)
mi. to nmi	Function (linked ; monadic)	mi.→nmi		Convert mile to nautical mile (nmi)
mi/kWh to kWh/100mi	Function (linked ; monadic)	mi/kWh→kWh/100mi		Convert mile per kilowatt-hour to kilowatt-hour per 100 mile (kWh/100mi)
mi <sup>2</sup> to km <sup>2</sup>	Function (linked ; monadic)	mi <sup>2</sup> →km <sup>2</sup>		Convert square mile to square kilometer (km <sup>2</sup> )
micro	Command (nonpgm)	μ		Factor 10 <sup>-6</sup> (181)
Micro sign	Character	μ		Character μ (181)
Middle dot	Character	·		Character · (183)
Mileage conversion	MENU	Ymmv:	Ymmv:	Convert between units of mileage ("Your mileage may vary") ("E" designates "kWh")
milli	Command (nonpgm)	·m		Factor 10 <sup>-3</sup>
Minima	Command	x <sub>MIN</sub>	x <sub>MIN</sub>	Minima (x <sub>MIN</sub> ; y <sub>MIN</sub> = (2 stack levels))
Minimum	Function (dyadic)	min	min	Minimum of X and Y
Minus	Function (dyadic)	-		Minus (45)
Minus infinity	Constant (#76)	-∞		inf.minus = -∞
Minus-plus	Character	∓		Character ∓ (8723)
Minutes	Function (monadic)	MIN	MIN	Minutes (of time)
Minutes & seconds	Function (cyclic ; monadic)	.ms	.ms	Convert sexagesimal format input sequence or decimal stack value to hh:mm:ss hours or dd°mm'ss" degrees (cyclic) (NIM input treated as sexagesimal (hh/dd.mmss) format ; stack input treated as decimal value)
Mirror bits	Function (monadic)	MIRROR	MIRROR	Flip bits
Miscellaneous conversions	MENU	Misc:	Misc:	Time, temperature, torque, power and field ratio conversions
ml to floz <sub>UK</sub>	Function (linked ; monadic)	ml→floz <sub>UK</sub>		Convert milliliter to UK fluid ounce (floz <sub>UK</sub> )
ml to floz <sub>US</sub>	Function (linked ; monadic)	ml→floz <sub>US</sub>		Convert milliliter to US fluid ounce (floz <sub>US</sub> )
mm to in.	Function (linked ; monadic)	mm→in.		Convert millimeter to inch (in.)
mm to point	Function (linked ; monadic)	mm→point		Convert millimeter to point (point)
mm.Hg to Pa	Function (linked ; monadic)	mm.Hg→Pa		Convert millimeter of Mercury to Pascal (Pa)
Mode settings	MENU	MODE	MODE	System (mode) settings with status indication and modification
Model	MENU	MODEL	MODEL	Functions to set parameter for which models are selected in the LR analysis
Modulo	Function (dyadic)	MOD	MOD	Y modulo X
Molar gas constant	Constant (#38)	R		c.mol.gas R = +8.31446261815324 (J/mol K)
Month	Function (monadic)	MONTH	MONTH	Month (of date)
Month day year	Setting (pgm)	MDY	MDY ( )	Date display mode MM/DD/YYYY (MM/DD/YYYY)
Month day year	Flag (system)	MDY		Date display mode MM/DD/YYYY (MM/DD/YYYY)
Moon orbit (semi major axis)	Constant (#02)	a <sub>Moon</sub>		orb.moon a <sub>Moon</sub> = +3.844 × 10 <sup>8</sup> (m)
More trig/hyperbolics	MENU (47)	TRG...	TRG...	Extended trigonometry (and access to hyperbolic) functions
mpg <sub>UK</sub> to l/100km	Function (linked ; monadic)	mpg <sub>UK</sub> →l/100km		Convert mile per gallon UK to liter per 100 km (l/100km)
mpg <sub>US</sub> to l/100km	Function (linked ; monadic)	mpg <sub>US</sub> →l/100km		Convert mile per gallon US to liter per 100 km (l/100km)
mpge <sub>UK</sub> to kWh/100mi	Function (linked ; monadic)	mpge <sub>UK</sub> →kWh/100mi		Convert mpg gasoline equivalent UK to kilowatt-hour per 100 mile (kWh/100mi)
mpge <sub>US</sub> to kWh/100mi	Function (linked ; monadic)	mpge <sub>US</sub> →kWh/100mi		Convert mpg gasoline equivalent US to kilowatt-hour per 100 mile (kWh/100mi)
mph to fur/ftn	Function (linked ; monadic)	mph→fur/ftn		Convert mile per hour to furlong per fortnight (fur/ftn)
mph to km/h	Function (linked ; monadic)	mph→km/h		Convert mile per hour to kilometer per hour (km/h)
mph to m/s	Function (linked ; monadic)	mph→m/s		Convert mile per hour to meter per second (m/s)
MU	Character	μ		Character μ (924)
mu lowercase	Character	μ		Character μ (956)
mū to m <sup>2</sup>	Function (linked ; monadic)	mū→m <sup>2</sup>		Convert mū to square meter (m <sup>2</sup> )
Much greater than	Character	»		Character » (8811)
Much less than	Character	«		Character « (8810)

FullName	Type	Label	Catalog	Description
Multiplication symbol	Flag	MULT*		Set for multiplication symbol * for exponential and complex notation, clear for .
Multiplication symbol •	Setting (pgm)	MULT•	MULT• ( )	Multiplication symbol • for exponential (and for complex numbers if CPXmul is set)
Multiplication symbol ×	Setting (pgm)	MULT×	MULT× (•)	Multiplication symbol × for exponential (and for complex numbers if CPXmul is set)
Multiply	Function (dyadic)	x	x	Multiply Y by X (215)
Multiply into	Function (monadic)	STO×	STO×	Multiply register or variable by X (STO× __ TamStoRcl menu)
Muon magnetic moment	Constant (#71)	$\mu_{\mu}$		mgmom.muon $\mu_{\mu} = -4.4904483 \times 10^{-26}$ (J/T)
Muon rest mass	Constant (#32)	$m_{\mu}$		mass.muon $m_{\mu} = +1.883531627 \times 10^{-28}$ (kg)
MVAR menu (internal)	MENU (internal)	MVAR (internal)		MVAR menu (internal) for VarMNU, integrator or solver
MyAlpha	MENU	My $\alpha$	My $\alpha$	User menu to quickly access user selected special characters ; all buttons are user assignable ; initially unpopulated ; reset using My $\alpha$ -R (alpha long [EXIT])
MyMenu	MENU	MyM	MyMenu	User menu presenting a ribbon to quickly access user selected functions (and menus) ; all buttons are user assignable (paneled look) ; set by RIBBONS ; reset using MyM-R ; default ribbon M.C47 (long [EXIT] ; Info : ribbons are documented as pages 1 : M.CPX ; 2 : M.FIN ; 3 : M.SAV ; 4 : M.C47 ; 5 : M.R47)
MyMenu	MENU (item)	MyMenu		User menu presenting a ribbon to quickly access user selected functions (and menus) ; all buttons are user assignable (paneled look) ; set by RIBBONS ; reset using MyM-R ; default ribbon M.C47 (long [EXIT])
MyMenu C47	Command	M.C47	M.C47	MyMenu primary F-key C47 ribbon ; leaves shifted rows intact ; documented as MyMenu page 4 (MyMenu primary F-key C47 ribbon)
MyMenu complex	Command	M.CPX	M.CPX	MyMenu primary F-key complex ribbon ; leaves shifted rows intact ; documented as MyMenu page 1 (MyMenu primary F-key complex ribbon)
MyMenu fff shortcut	Setting	MyM.3	MyM.3 [ ]	MyMenu activated by triple (or longer) shift ([f/g]) (MyM.3 and HOME.3 are linked (both cannot be active) ; functional in 1-shift layouts)
MyMenu financial	Command	M.FIN	M.FIN	MyMenu primary F-key financial ribbon ; leaves shifted rows intact ; documented as MyMenu page 2 (MyMenu primary F-key financial ribbon)
MyMenu R47	Command	M.R47	M.R47	MyMenu primary F-key R47 ribbon ; leaves shifted rows intact ; documented as MyMenu page 5 (MyMenu primary F-key R47 ribbon)
MyMenu reset	Command	MyM-R		Reset all MyMenu user assignments (C47 MyMenu menu cleaned)
MyMenu save/load	Command	M.SAV	M.SAV	MyMenu primary F-key save/load ribbon ; leaves shifted rows intact ; documented as MyMenu page 3 (MyMenu primary F-key save/load ribbon)
MyMenu shown	Setting	MyMb	MyMb [•]	Base MyMenu shown (when all menus are exited using EXIT) (MyMb and HOMEb are linked (both cannot be active))
My $\alpha$ menu reset	Command	My $\alpha$ -R		Reset all My $\alpha$ menu user assignments (C47 My $\alpha$ menu cleaned)
N	Character	N		Character N (78)
n	Command	n	n $\Sigma$	Number of samples
N acute	Character	Ñ		Character Ñ (323)
n acute lowercase	Character	ñ		Character ñ (324)
N caron	Character	Ň		Character Ň (327)
n caron lowercase	Character	ň		Character ň (328)
n lowercase	Character	n		Character n (110)
N tilde	Character	Ñ		Character Ñ (209)
n tilde lowercase	Character	ñ		Character ñ (241)
N to lbf	Function (linked ; monadic)	N→lbf		Convert Newton to poundforce (lbf:)
N47 layout	Layout (SIM)	N47		N47 keyboard layout ; this layout works in USER mode on the simulator (N47: Exp 2 shifts L (32 mould) /x-+ R ↕ top)
Nabla	Character	∇		Character ∇ (8711)
Nand	Character	ā		Character ā (8892)
nano	Command (nonpgm)	-n		Factor 10 <sup>-9</sup>
Narrow space	Character	u		Character u (826)

FullName	Type	Label	Catalog	Description
N-ary product	Character	$\prod$		Character $\prod$ (8719)
N-ary product	Character	$\prod$		Character $\prod$ (8719)
N-ary sum	Character	$\sum$		Character $\sum$ (8721)
Natural logarithm	Function (monadic)	LN	LN	Natural logarithm (base $e$ )
Natural logarithm	Symbol	LN		Natural logarithm (base $e$ ) (alpha g [LN] ("LN"))
Natural N	Character	N		Character N (8469)
Negative binomial (quantile)	Function (monadic)	NBin <sup>-1</sup>	NBin <sup>-1</sup>	Negative binomial inverse cumulative distribution
Negative binomial cdf (lower)	Function (monadic)	NBin <sub>l</sub>	NBin <sub>l</sub>	Negative binomial cumulative distribution (lower tail)
Negative binomial cdf (lower)	Function (monadic)	NBin <sub>l</sub>	NBin <sub>l</sub>	Negative binomial cumulative distribution (lower tail)
Negative binomial pmf	Function (monadic)	NBin <sub>p</sub>	NBin <sub>p</sub>	Negative binomial probability mass function
Negative exclamation mark	Character	Ⓜ		Character Ⓜ (9263)
Neighbour?	Function (dyadic)	NEIGHB	NEIGHB	Get neighbour value of X compared to Y (next integer or next machine representable real)
Neutron / proton rest mass	Constant (#26)	$m_n/m_p$		r.neu.prot $m_n/m_p = +1.00137841898$
Neutron magnetic moment	Constant (#68)	$\mu_n$		magnom.neu $\mu_n = -9.662365 \times 10^{-27}$ (J/T)
Neutron rest mass	Constant (#25)	$m_n$		mass.neu $m_n = +1.67492749804 \times 10^{-27}$ (kg)
New equation	MENU (item)	NEW	EQ.NEW	Open equation editor menu to create new equation (previous equation pushed) (EIM = Equation Input Mode ; Ref : Equation editor)
New matrix	Command (submnu)	NEW	M.NEW	Create new matrix in X
New program	Command	X.NEW		New XEQC program
Newtonian constant of gravitation	Constant (#12)	G		c.grav.nwt $G = +6.6743 \times 10^{-11}$ (m <sup>3</sup> /kg.s <sup>2</sup> )
Next fit	Command (cyclic ; nonpgm)	NXTFIT		Assess next curve fitting (cyclic)
Next prime	Function (monadic)	NEXTP	NEXTP	Next prime number (Prime numbers are calculated only if $X < 10^{308}$ , or an error is returned ("An argument exceeds the function domain"))
Nm to lbf-ft	Function (linked ; monadic)	Nm→lbf-ft		Convert Newton-meter to pound-foot (lbf-ft:)
nmi to km	Function (linked ; monadic)	nmi→km		Convert nautical mile to kilometer (km:)
nmi to m	Function (linked ; monadic)	nmi→m		Convert nautical mile to meter (m:)
nmi to mi.	Function (linked ; monadic)	nmi→mi.		Convert nautical mile to mile (mi.:)
nmi <sup>2</sup> to km <sup>2</sup>	Function (linked ; monadic)	nmi <sup>2</sup> →km <sup>2</sup>		Convert square nautical mile to square kilometer (km <sup>2</sup> :)
No	Command	NO		Respond : No ([EXIT])
No operation	Command (PEM)	NOP	NOP	No operation (empty step)
Nominal mean angular velocity of the Earth	Constant (#75)	$\omega_\oplus$		vangl.earth $\omega_\oplus = +7.292115 \times 10^{-5}$ (rad/s)
NOR	Function (dyadic)	NOR	NOR	Logical not OR (bitwise)
Nor	Character	$\bar{v}$		Character $\bar{v}$ (8893)
Normal & Log normal distribution	MENU	Norml:	Norml:	Normal & Log normal probability distribution (continuous) (RegM = $\mu$ = mean ; RegS = $\sigma$ = standard deviation)
Normal (quantile)	Function (monadic)	Norml <sup>-1</sup>	Norml <sup>-1</sup>	Normal inverse cumulative distribution
Normal cdf (lower)	Function (monadic)	Norml <sub>l</sub>	Norml <sub>l</sub>	Normal cumulative distribution (lower tail)
Normal cdf (upper)	Function (monadic)	Norml <sub>u</sub>	Norml <sub>u</sub>	Normal cumulative distribution (upper tail)
Normal pdf	Function (monadic)	Norml <sub>p</sub>	Norml <sub>p</sub>	Normal probability density function
Normal results	Command	SPCRESO	SPCRESO	Do not allow special results of calculations (infinity, not-a-number) ; an error will occur for such events (For programming purposes)
Normal-super-subscript	Setting (cyclic)	<sup>SUP</sup> <sub>SUB</sub>		Cycle normal-super-subscript (locking) ( <sup>SUP</sup> <sub>SUB</sub> ; $\alpha^{SUP}$ ; $\alpha_{SUB}$ )
NOT	Function (monadic)	NOT	NOT	Logical NOT
Not	Character	$\neg$		Character $\neg$ (172)
Not a Number	Constant (#36)	NaN		not.a.nr NaN = Not a number
Not a number?	Function (monadic)	NaN?	NaN?	Test X is Not-a-Number
Not A Subset Of	Character	$\not\subseteq$		Character $\not\subseteq$ (8836)
Not an element of	Character	$\notin$		Character $\notin$ (8713)
Not AND	Function (dyadic)	NAND	NAND	Logical not AND (bitwise)
Not equal to	Character	$\neq$		Character $\neq$ (8800)

FullName	Type	Label	Catalog	Description
Not parallel to	Character	∥		Character ∥ (8742)
NU	Character	N		Character N (925)
nu lowercase	Character	ν		Character ν (957)
Nuclear magneton	Constant (#70)	$\mu_{\nu}$		magn.nucl $\mu_{\nu} = +5.0507837461 \times 10^{-27}$ (J/T)
Number (base)	Setting (pgm ; stack)	#	→INT	Set number base ; operates on all stack registers depending on BASE <sub>HP</sub> ; reset by [.d] (g [LOG]) (Indirection (→) activates TamNonRegInd ; #TAM shortcuts : B = BIN ; D = DEC ; ENTER = DEC ; H = HEX ; O = OCT ; Info : SBI depends on SBfrac)
Number (base)	Setting (pgm ; stack)	→INT	→INT	Set number base ; operates on all stack registers depending on BASE <sub>HP</sub> ; reset by [.d] (g [LOG]) (Indirection (→) activates TamNonRegInd ; #TAM shortcuts : B = BIN ; D = DEC ; ENTER = DEC ; H = HEX ; O = OCT ; Info : SBI depends on SBfrac)
Number base	MENU	BASE	BASE	Number base operations (shortint) (X: hexadecimal ; X: binary)
Number of bins	Command	nBINS		Number of bins in histogram (nBINS : ; ↓BIN : ; ↑BIN : (3 stack levels))
Number of bits set	Function (monadic)	#B	#B	Count number of bits set
Number of local registers?	Command	LocR?	LocR?	Get number of local registers (current routine) (Set by LocR)
Number of payments	Variable (real)	n	NPPER	Total number of payments (NPPER = n pay periods = )
Number sign	Symbol	#		Character # (alpha f[.] ; Code : 35)
Number variables	MENU	NUMBRS	NUMBRS	Auto-generated catalog of variables of the specified type : number
Numeric entry	Flag	NUM.IN		Numeric entry active
Numlock	Setting	NUM		Lock numeric alpha input
0	Character	0		Character 0 (79)
0 acute	Character	0̇		Character 0̇ (211)
o acute lowercase	Character	ó		Character ó (243)
0 breve	Character	0̂		Character 0̂ (334)
o breve lowercase	Character	ö		Character ö (335)
0 circumflex	Character	0̂		Character 0̂ (212)
o circumflex lowercase	Character	ô		Character ô (244)
0 diaeresis	Character	ö		Character ö (214)
o diaeresis lowercase	Character	ö		Character ö (246)
0 grave	Character	0̀		Character 0̀ (210)
o grave lowercase	Character	ò		Character ò (242)
o lowercase	Character	o		Character o (111)
0 macron	Character	0̄		Character 0̄ (332)
o macron lowercase	Character	ō		Character ō (333)
0 stroke	Character	0̸		Character 0̸ (216)
o stroke lowercase	Character	ø		Character ø (248)
0 tilde	Character	0̃		Character 0̃ (213)
o tilde lowercase	Character	õ		Character õ (245)
Obelus	Character	÷		Character ÷ (247)
Oblique 1	Character	↙		Character ↙ (9249)
Oblique 2	Character	↘		Character ↘ (9250)
Oblique 3	Character	↗		Character ↗ (9253)
Oblique 4	Character	↖		Character ↖ (9254)
Octal	Setting (pgm)	OCT	OCT	Convert X to octal and toggle octal mode (#TAM [EEX] (0))
Odd?	Function (monadic)	ODD?	ODD?	Test X is integer AND odd
OE	Character	œ		Character œ (338)
oe lowercase	Character	œ		Character œ (339)
Off	Command	OFF	OFF	Turn off calculator
Old (matrix)	Command	OLD	M.OLD	Revert to old element (while editing)
OMEGA	Character	Ω		Character Ω (937)

FullName	Type	Label	Catalog	Description
omega lowercase	Character	ω		Character ω (969)
omega tonos lowercase	Character	ώ		Character ώ (974)
OMICRON	Character	0		Character 0 (927)
omicron lowercase	Character	o		Character o (959)
omicron tonos lowercase	Character	ó		Character ó (972)
OMp209	User Program	OMp209		OMp209 test program, preloaded from testPgms.bin
OMp212	User Program	OMp212		OMp212 test program, preloaded from testPgms.bin
OMp224	User Program	OMp224		OMp224 test program, preloaded from testPgms.bin
OMp226	User Program	OMp226		OMp226 test program, preloaded from testPgms.bin
OMp233	User Program	OMp233		OMp233 test program, preloaded from testPgms.bin
OMp235	User Program	OMp235		OMp235 test program, preloaded from testPgms.bin
OMp237	User Program	OMp237		OMp237 test program, preloaded from testPgms.bin
op_i	Character	i		Character i (<none> ; Code : 8520)
op_j	Character	j		Character j (<none> ; Code : 8521)
Operator a	Command	a	op_a	Insert value of 1 ∠ 120°
Operator a²	Command	a²	op_a²	Insert value of 1 ∠ 240°
OR	Function (dyadic)	OR	OR	Logical OR
Or	Character	ν		Character ν (8744)
Orthogonal	MENU	Orthog	ORTHOg	Orthogonal polynomials
Orthogonal fit	Setting	OrthoF	OrthoF [ ]	Select orthogonal curve fitting (used in SCATR) ; deselects other curve fit settings (Sets BestF to 0 ; TI : L.R. selected to OrthoF)
Osci	User Program	Osci		Osci test program, preloaded from testPgms.bin
Overflow	Flag	OVERFL		Status of overflow bit (SBI depends on SBoc)
Overflow Carry	Character	ē		Character ē (169)
oz to g	Function (linked ; monadic)	oz→g		Convert ounce to gram (g:)
P	Character	P		Character P (80)
p lowercase	Character	p		Character p (112)
P.FN menu reset	Command	P.FN-R		Reset all P.FN user assignments (C47 P.FN menu reset to default)
Pa to atm	Function (linked ; monadic)	Pa→atm		Convert Pascal to atmosphere (atm:)
Pa to bar	Function (linked ; monadic)	Pa→bar		Convert Pascal to bar (bar:)
Pa to in.Hg	Function (linked ; monadic)	Pa→in.Hg		Convert Pascal to inch of Mercury (in.Hg:)
Pa to mm.Hg	Function (linked ; monadic)	Pa→mm.Hg		Convert Pascal to millimeter of Mercury (mm.Hg:)
Pa to psi	Function (linked ; monadic)	Pa→psi		Convert Pascal to pounds per square inch (psi:)
Pa to torr	Function (linked ; monadic)	Pa→torr		Convert Pascal to torr (torr:)
Pair of parentheses	Character	( )		Characters ( )
Parabolic fit	Setting	ParabF	ParabF [ ]	Set BestF to include parabolic curve fitting ; deselects OrthoF (Adds 64 to BestF ; TI : L.R. selected to <BestF>)
Parallel	Function (dyadic)			Parallel impedance = (X × Y) / (X + Y) (8741)
Parallel to	Character			Character    (8741)
Partial differential	Character	∂		Character ∂ (8706)
Pause	Command (PEM)	PAUSE	PAUSE	Pause program for n ticks (0-99 ; one tick is 10 ms) ; continues after delay or on keypress (PAUSE __ TamNonReg menu)
Payment	Variable (real)	PMT	PMT	Payment (reserved real variable) ; displays as P in menu TVM when value is large (PMT = Payment = )
Payment period	Variable (real)	pp/a	PPER/a	Payment period per annum (reserved real variable) (PPER/a = Pay periods p.a. = )
pc to m	Function (linked ; monadic)	pc→m		Convert parsec to meter (m)
Pct of sum and Delta pct to mean	Function (monadic)	%Σ,Δ% $\bar{x}$	%Σ,Δ% $\bar{x}$	Percentage of x to Σx and Delta percentage to mean using statistics matrix (STATS) (% ; Δ% : (2 stack levels))
Percent	Function (dyadic)	%	%	X Percent of Y, keeping Y on stack
Percent	Character	%		Character % (37)



FullName	Type	Label	Catalog	Description
Percentage of sum	Function (monadic)	%Σ	%Σ	Percentage of x to Σx (% :)
Percentage of total	Function (dyadic)	%T	%T	Percentage of total, keeping Y on stack (% :)
Percentile	Command	x%ILE	x%ILE	Percentile (pctile <sub>x</sub> ; pctile <sub>y</sub> = (2 stack levels))
Permutations	Function (dyadic)	P <sub>yx</sub>	PERM	Permutations of X out of Y
Perpendicular	Character	⊥		Character ⊥ (8869)
Peta	Command (nonpgm)	-P		Factor 10 <sup>15</sup>
Peta binary	Command (nonpgm)	-Pi		Factor 2 <sup>50</sup>
PFall	User Program	PFall		PFall test program, preloaded from testPgms.bin
Pgm begin	Character	⌈		Character ⌈ (9259)
PHI	Character	φ		Character φ (934)
phi lowercase	Character	φ		Character φ (966)
pi	Command	π		Value of pi (π)
PI	Character	Π		Character Π (928)
pi lowercase	Symbol	π		Greek character π (alpha g[R+]; Code : 960)
pico	Command (nonpgm)	-p		Factor 10 <sup>-12</sup>
PICTURE	User Program	PICTURE		PICTURE test program, preloaded from testPgms.bin
Pipe	Character			Character   (124)
Planck	Character	h		Character h (8462)
Planck / 2pi	Character	ħ		Character ħ (8463)
Planck constant	Constant (#18)	h		c.planck h = +6.62607015 × 10 <sup>-34</sup> (Js)
Planck length	Constant (#22)	l <sub>PL</sub>		len.planck l <sub>PL</sub> = +1.616255 × 10 <sup>-35</sup> (m)
Planck mass	Constant (#28)	m <sub>PL</sub>		mass.planck m <sub>PL</sub> = +2.176435 × 10 <sup>-8</sup> (kg)
Planck temperature	Constant (#51)	T <sub>P</sub>		temp.planck T <sub>P</sub> = +1.416785 × 10 <sup>32</sup> (K)
Planck time	Constant (#52)	t <sub>PL</sub>		time.planck t <sub>PL</sub> = +5.391245 × 10 <sup>-44</sup> (s)
Play sounds	Command (HW)	PLAY	PLAY	Play sounds (input from n×3 or n×2 matrix variable having rows : [frequency, duration, volume] with frequency in Hz (0 = silent) ; duration in ms (max 2000) ; volume) ; volume element is optional) ; hardware only (PLAY _ TamLabel menu)
Plot statistics	MENU (item)	PLSTAT	PLSTAT	Plot statistics matrix (STATS) (CAT.MENU)
Plot statistics	MENU	PLSTMNU		Plot statistics matrix (STATS)
Plotting	MENU	PLOT	PLOT	Plotting and summation functions
Plus	Function (dyadic)	+		Plus (43)
Plus infinity	Constant (#77)	+∞		inf.plus = +∞
Plus-minus	Character	±		Character ± (alpha g[CHS] ; Code : 177)
point to mm	Function (linked ; monadic)	point→mm		Convert point to millimeter (mm)
Poisson (quantile)	Function (monadic)	Poiss <sup>-1</sup>	Poiss <sup>-1</sup>	Poisson inverse cumulative distribution
Poisson (quantile)	Function (monadic)	Poiss <sub>p</sub>	Poiss <sub>p</sub>	Poisson inverse cumulative distribution
Poisson cdf (lower)	Function (monadic)	Poiss <sub>Λ</sub>	Poiss <sub>Λ</sub>	Poisson cumulative distribution (lower tail)
Poisson cdf (upper)	Function (monadic)	Poiss <sub>Δ</sub>	Poiss <sub>Δ</sub>	Poisson cumulative distribution (upper tail)
Poisson distribution	MENU	Poiss:	Poiss:	Poisson probability distribution (discrete) (RegR = λ = expected rate of events)
Polar	Setting (pgm)	POLAR	POLAR ( )	Polar representation of complex numbers (internal value is RECT) (SBI depends on SBcpX)
Polar	Flag	POLAR		Set for polar representation of complex numbers, clear for rectangular display (SBI depends on SBcpX)
Polar (Character)	Character	⊙		Character ⊙ (8857)
Polynomial 4th degree	Symbol	Poly4		Create polynomial : b4 × x <sup>4</sup> + b3 × x <sup>3</sup> + b2 × x <sup>2</sup> + b1 × x + b0
Pop local registers	Command (PEM)	PopLR	PopLR	Pop local registers (no return to calling routine)
Population standard deviation	Command	σ	σ	Population standard deviation (σ <sub>x</sub> ; σ <sub>y</sub> = (2 stack levels))
Pound	Character	£		Character £ (163)
Power conversion	MENU	Power:	Power:	Convert between units of power

FullName	Type	Label	Catalog	Description
Power fit	Setting	PowerF	PowerF [ ]	Set BestF to include power curve fitting ; deselects OrthoF (Adds 8 to BestF ; TI : L.R. selected to <BestF>)
Precision	Command	S <sub>mi</sub>	S <sub>mi</sub>	Precision of measuring instrument investigated, requires 30 data pairs (s <sub>mi</sub> =)
Prefixes (UNIT)	MENU	Prefix	Prefix	Prefixes for UNIT mode
Present value	Variable (real)	PV	PV	Present value (reserved real variable) ; displays as P in menu TVM when value is large (PV = Present Value = )
Prime	User Program	Prime		Prime test program, preloaded from testPgms.bin
Prime factors	Function (monadic)	FACTORS	FACTORS	Prime factors of the integer input in X ; factors are calculated in longint, stored in real type (34 digits) and returned as a 2 row matrix (row 1 contains the primes, row 2 contains the powers) ; with iteration counter ; interrupt by keypress (Prime factors will be accurate up to 3 317 044 064 679 887 385 961 980)
Prime?	Function (monadic)	PRIME?	PRIME?	Test X is prime (The prime test (PRIME?) is correct up to 3 317 044 064 679 887 385 961 980 ; above that limit there are no false negatives, but there is a very low probability of false positives (albeit no known cases))
Print all registers	Command	▣ALLr	▣ALLr	Print all registers (Outputs r <tab> 1.234 <newline> for all registers)
Print byte	Command (strike)	▣*	▣*	Print single byte
Print character	Command (strike)	▣CHAR	▣CHAR	Print single character
Print column	Command (strike)	▣TAB	▣TAB	Position print head to column
Print LCD	Command (strike)	▣LCD	▣LCD	Print screen contents
Print linefeed	Command (strike)	▣ADV	▣ADV	Print linefeed (advance)
Print program	Command	▣PROG	▣PROG	Print program (▣PROG _ TamLabel ; DMCP : File save dialog (PROGRAMS/))
Print register	Command	▣r	▣r	Print register r (▣r _ Tam menu ; Info : Outputs r <tab> 1.234 <newline>)
Print registers	Command (strike)	▣REGS	▣REGS	Print registers (sss.nn means print registers from sss through sss + nn - 1)
Print stack	Command	▣STK	▣STK	Print stack registers
Print sums	Command (strike)	▣Σ	▣Σ	Print summation registers
Print user items	Command (strike)	▣USER	▣USER	Print user items (variable names and programs)
Print X	Command	▣x	▣x	Print stack register X (Outputs X <tab> 1.234 <newline>)
Print X and Y	Command	▣xy	▣xy	Print stack registers X and Y (Outputs X <tab> 1.234 <tab> Y <tab> 1.234 <newline>)
Printer	Character	▣		Character ▣ (9113)
Printer Active	Flag	PRTACT		Printing is enabled
Printer delay	Setting (strike)	▣DLAY	▣DLAY	Set printer delay
Printer mode	Setting (strike)	▣MODE	▣MODE	Set printer mode (0 : printer font ; 1 : variable ; 2 : small ; 3 : ASCII)
Printer off	Command	PROFF	PROFF	Set printer off
Printer on	Command	PRON	PRON	Set printer on
Printer on/off	Setting	PRNTR	PRNTR [ ]	Set printer on/off
Printer width	Setting (strike)	▣WIDTH	▣WIDTH	Set number of print columns or pixels (depending on ▣MODE)
Printing	MENU	PRINT	PRINT	Printing functions (Print commands append data to file DATA/<YYYYMMDD>-HHMMSS00>REGS.TSV in FAT ; new datafile after timeout of 2 minutes)
Printing	Flag (system)	PRINTS		Calculator is sending data to printer (SBI depends on SBprn)
Probability	MENU	PROB	PROB	Probability functions
PROD	User Program	PROD		PROD test program, preloaded from testPgms.bin
Product (programmable)	Command	Π <sub>n</sub>	Π <sub>n</sub>	Real or complex product using specified program ; interrupt by keypress ; if MONIT is set, displays current product and iteration counter (Π <sub>n</sub> _ TamLabel menu ; <from> ENTER <to> ENTER <step>)
PROG	MENU (TAM ; ASM)	PROG	PROG	Presented in TAM menus for commands accessing labels (CAT.PROGS.* menu ; Type characters 1-2 to search)
Programming	Command	PRGM		Enter programming mode (PEM = Program Entry Mode ; starts UPPERCASE)

FullName	Type	Label	Catalog	Description
Programming functions	MENU	P.FN	P.FN	User menu to quickly access selected menus and functions for programming ; all buttons are user assignable (paneled look) ; initially populated for basic programming options ; reset using P.FN-R
Programming functions 1	MENU	P.FN1	P.FN1	Programming functions 1 (fixed)
Programming functions 2	MENU	P.FN2	P.FN2	Programming functions 2 (fixed)
Programs	MENU (ASM)	PROGS	PROGS	Auto-generated catalog of programs
Proper Fractions	Flag	PROPFR		Proper fractions are used (or improper)
Proper Fractions	Setting	PROPFR	PROPFR [.]	Proper fractions are used (or improper)
Proportional to	Character	$\alpha$		Character $\alpha$ (8733)
Proton / electron rest mass	Constant (#29)	$m_p/m_e$		r.prot.elec $m_p/m_e = +1.83615267343 \times 10^3$
Proton gyromagnetic ratio	Constant (#58)	$\gamma_p$		r.gyro.prot $\gamma_p = +2.6752218744 \times 10^8$
Proton magnetic moment	Constant (#69)	$\mu_p$		mgmom.prot $\mu_p = +1.41060679736 \times 10^{-26}$ (J/T)
Proton rest mass	Constant (#27)	$m_p$		mass.prot $m_p = +1.67262192369 \times 10^{-27}$ (kg)
PSI	Character	$\psi$		Character $\psi$ (936)
psi lowercase	Character	$\psi$		Character $\psi$ (968)
psi to Pa	Function (linked ; monadic)	psi→Pa		Convert pounds per square inch to Pascal (Pa:)
Put keycode in buffer	Command	PUTK	PUTK	Copy keycode from register to keyboard buffer for immediate execution (PUTK ___ Tam menu)
Put submatrix	Command	PUTM	M.PUT	Put submatrix in X into the indexed matrix starting at current element
pwr to dB	Function (linked ; monadic)	pwr→dB		Convert power ratio to decibel (dB:)
Q	Character	Q		Character Q (81)
q lowercase	Character	q		Character q (113)
QOPPA	Character	Q		Character Q (984)
qoppa lowercase	Character	q		Character q (985)
QR decomposition	Function (monadic)	M.QR	M.QR	Decompose matrix in X into unitary (Q) in Y and upper (R) matrix in X (QR factorisation)
qt. to l	Function (linked ; monadic)	qt.→l		Convert quart to liter (l:)
Quadratic means	Command	$\bar{x}_{RMS}$	$\bar{x}_{RMS}$	Quadratic means (root mean square) ( $\bar{x}_{RMS}$ ; $\bar{x}_{RMS} = (2$ stack levels))
Question mark	Character	?		Character ? (63)
Quiet	Flag	QUIET		Beeper is disabled
Quotation mark	Character	"		Character " (34)
R	Character	R		Character R (82)
R acute	Character	$\acute{R}$		Character $\acute{R}$ (340)
r acute lowercase	Character	$\acute{r}$		Character $\acute{r}$ (341)
R caron	Character	$\check{R}$		Character $\check{R}$ (344)
r caron lowercase	Character	$\check{r}$		Character $\check{r}$ (345)
r lowercase	Character	r		Character r (114)
R47 layout	Layout	R47		R47 keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS-R) (R47: 2 shifts R (43S mould) /x++ R)
R47bkfg layout	Layout	R47bkfg		R47bkfg keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS-R) (R47bkfg: 1 shift R (43Ssp mould) /x++ R)
R47fg_g layout	Layout	R47fg_g		R47fg_g keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS-R) (R47fg_g: 2 shifts f/g (43Ssp mould) /x++ R)
R47fgbk layout	Layout	R47fgbk		R47fgbk keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS-R) (R47fgbk: 1 shift L (43Ssp mould) /x++ R)
rad to deg	Function (linked ; monadic)	rad→deg		Convert radian to degree (untagged) (deg:)
rad to grad	Function (linked ; monadic)	rad→grad		Convert radian to gradian (untagged) (grad:)
rad/s to RPM	Function (linked ; monadic)	rad/s→RPM		Convert radian per second to rotation per minute (RPM:)
Radix	Symbol	.		Enter radix (default ".") (Second press enters fraction mode)
Radix comma	Setting (pgm)	COM,	RCOM, ( )	Radix decimal comma
Radix dot	Setting (pgm)	DOT.	RDOT. ( )	Radix decimal dot

FullName	Type	Label	Catalog	Description
Radix period	Setting (pgm)	PER.	RPER. (•)	Radix decimal period
Radix wide comma	Setting (pgm)	WCOM,	RWCOM, ( )	Radix decimal wide comma
Radix wide dot	Setting (pgm)	WDOT •	RWDOT • ( )	Radix decimal wide dot
Radix wide period	Setting (pgm)	WPER.	RWPER. ( )	Radix decimal wide period
Random integer	Function (dyadic)	RANI#	RANI#	Random integer : lower ENTER upper ; keeping input on stack (X: result, Y: upper, Z: lower)
Random number	Command	RAN#	RAN#	Random number (real)
Range	Setting	RNG	RNG <sub>6,145</sub>	Set range as $10^{nn}$ ; $99 \leq nn \leq 6145$ (RNG ____ TamNonReg menu)
Range?	Command	RANGE?	RANGE?	Get range as $10^{nn}$ (Set by RNG)
Ratio	Character	:		Character : (8758)
Rational Q	Character	Q		Character Q (8474)
Read program	Command	READP	READP	Read program from WRITEP file PROGRAMS/<program>.p47 in FAT (Program file loaded ; DMCP File open dialog (PROGRAMS/))
Real	MENU	REAL	REAL	Functions on real and complex numbers
Real number display format	Variable (longint)	REALDF	REALDF	Real number display format (reserved long integer variable, write protected) (ALL: 0 ; FIX: 1 ; SCI: 2 ; ENG: 3 ; SIG: 4 ; UNIT: 5)
Real one	Command (TAM)	1.		Presented in TamCmp menu for comparisons (e.g. x< ?) to quickly enter 1.
Real part	Function (monadic)	Re	Re	Real part of complex number
Real R	Character	R		Character R (8477)
Real results	Command	CPXRES0	CPXRES0	Do not allow complex results for real input ; an error will occur for such events (For programming purposes)
Real solver (stack limits)	Command (nonpgm)	realSlv <sub>x</sub>	realSlv <sub>x</sub>	Solve the algebraic expression ( = 0 ) entered in NEW [EQN] for real roots ; interrupt by keypress ; tolerance set by SDIGS ; monitoring set by MONIT (Result Code = ; Accuracy ≈ ; <var> <sub>prev</sub> = ; <var> = ; (4 stack levels))
Real solver (variable limits)	Command (nonpgm)	realSlv	realSlv	Solve the algebraic expression ( = 0 ) entered in NEW [EQN] for real roots ; interrupt by keypress ; tolerance set by SDIGS ; monitoring set by MONIT (Result Code = ; Accuracy ≈ ; <var> <sub>prev</sub> = ; <var> = ; (4 stack levels))
Real to complex	Function (dyadic)	RE→CX	RE→CX	Convert reals to complex (in POLAR, using angle tag or ADM)
Real variables	MENU	REALS	REALS	Auto-generated catalog of variables of the specified type : real
Real zero	Command (TAM)	0.		Presented in TamCmp menu for comparisons (e.g. x< ?) to quickly enter 0.
Real?	Function (monadic)	REAL?	REAL?	Test X is real
Recall (add to timer)	Command (nonpgm)	RCL		Recall register value and add to running stopwatch timer (decimal hours)
Recall (register)	Command	RCL	RCL	Recall value from register or variable can be followed by +, -, x, ÷ for recall and add, recall and subtract, recall and multiply, recall and divide functions (RCL __ TamStoRcl menu)
Recall and add	Function (monadic)	RCL+	RCL+	Recall register or variable and add X (RCL+ __ TamStoRcl menu)
Recall and divide	Function (monadic)	RCL/	RCL/	Recall register or variable and divide by X (RCL/ __ TamStoRcl menu)
Recall and multiply	Function (monadic)	RCL×	RCL×	Recall register or variable and multiply by X (RCL× __ TamStoRcl menu)
Recall and subtract	Function (monadic)	RCL-	RCL-	Recall register or variable and subtract X (RCL- __ TamStoRcl menu)
Recall configuration	Command	RCLCFG	RCLCFG	Recall configuration from register or variable (RCLCFG __ TamStoRcl menu)
Recall current element	Command	RCLLEL	RCLLEL	Recall current element of indexed matrix ( $M[I, J_c] = M[\text{<row> <col>}]$ )
Recall current index	Command	RCLIJ	RCLIJ	Recall current index (row I and column J) ( $[I, J_c] = [\text{<row> <col>}]$ )
Recall maximum	Command	RCL↑	RCL↑	Recall maximum of X and register or variable (RCL↑ __ TamStoRcl menu)
Recall minimum	Command	RCL↓	RCL↓	Recall minimum of X and register or variable (RCL↓ __ TamStoRcl menu)
Recall next element	Command	RCLSEQ	RCLSEQ	Recall next element of indexed matrix (increase index and read element) (Increases row I and column J and accesses element)
Recall stack	Command	RCLS	RCLS	Recall complete stack from 4 or 8 registers (RCLS __ TamStoRcl menu)
Recall triple I	Command	RCL 3I	RCL 3I	Copy R96, R97, R98 to X, Y, Z
Recall triple V	Command	RCL 3V	RCL 3V	Copy R93, R94, R95 to X, Y, Z
Recall triple Z	Command	RCL 3Z	RCL 3Z	Copy R90, R91, R92 to X, Y, Z
Recall vector element	Command	RCLVEL	RCLVEL	Recall element n (TAM) of vector or matrix in X (RCLVEL __ Tam menu)

FullName	Type	Label	Catalog	Description
Reciprocal	Function (monadic)	$1/x$	$1/x$	Reciprocal ( $1/x$ ) (When X is a matrix $1/x$ inverts it ( $[M]^{-1}$ ))
Rectangular	Setting (pgm)	RECT	RECT (*)	Rectangular display of complex numbers (internal value is RECT) (SBI depends on SBcpX)
Reduced Planck constant	Constant (#19)	$\hbar$		red.planck $\hbar = +1.054571817646156391262428003302281 \times 10^{-34}$ (Js)
Register A	Shortcut (TAM)	A		Easy access to register A (R221) : Stack level 5 (STO/RCL {A} (TAM [Σ+]) ; TI : A:)
Register A	Variable (register)	A		Register A (reserved variable)
Register B	Shortcut (TAM)	B		Easy access to register B (R222) : Stack level 6 (STO/RCL {B} (TAM [ $1/x$ ]) ; TI : B:)
Register B	Variable (register)	B		Register B (reserved variable)
Register browser	Browser	REGS	REGS	Browse all registers ([.] : Switch register/variable view, [R/S] : Switch contents/storage view, [RCL] : Recall bottom item, [▲], [▼], A..D, I..L, 00..99 : Navigate)
Register C	Shortcut (TAM)	C		Easy access to register C (R223) : Stack level 7 (STO/RCL {C} (TAM [ $\sqrt{x}$ ]) ; TI : C:)
Register C	Variable (register)	C		Register C (reserved variable)
Register D	Shortcut (TAM)	D		Easy access to register D (R224) : Stack level 8 (STO/RCL {D} (TAM [LOG]) ; TI : D:)
Register D	Variable (register)	D		Register D (reserved variable)
Register E	Shortcut (TAM)	E		Easy access to register E (R200) : General purpose register E (STO/RCL {E} (TAM [LN]) ; TI : E:)
Register F	Shortcut (TAM)	F		Easy access to register F (R201) : General purpose register F (STO/RCL F ([REG.2] F2) ; TI : F:)
Register G	Shortcut (TAM)	G		Easy access to register G (R202) : General purpose register G (STO/RCL {G} (TAM [STO]) ; TI : G:)
Register H	Shortcut (TAM)	H		Easy access to register H (R203) : General purpose register H (STO/RCL {H} (TAM [RCL]) ; TI : H:)
Register I	Shortcut (TAM)	I		Easy access to register I (R204) : Matrix row index (STO/RCL {I} (TAM [R+]) ; TI : I:)
Register I	Variable (register)	I		Register I (reserved variable)
Register J	Shortcut (TAM)	J		Easy access to register J (R205) : Matrix column index (STO/RCL {J} (TAM [SIN]) ; TI : J:)
Register J	Variable (register)	J		Register J (reserved variable)
Register K	Shortcut (TAM)	K		Easy access to register K (R206) ; Used in MVAR and in the Real solver, stores variable name when <var...> button is pressed without NIM input (STO/RCL {K} (TAM [COS]) ; TI : K:)
Register K	Variable (register)	K		Register K (reserved variable)
Register L	Shortcut (TAM)	L		Easy access to register L (R225) : Last X (STO/RCL {L} (TAM [TAN]) ; TI : L:)
Register L	Variable (register)	L		Register L (reserved variable)
Register M	Shortcut (TAM)	M		Easy access to register M (R207) : Distribution parameter : mean/median/location/degree of freedom 1 (STO/RCL {M} (TAM [ $x\bar{y}$ ]) ; TI : M:)
Register N	Shortcut (TAM)	N		Easy access to register N (R208) : Distribution parameter : degree of freedom 2/quantity (STO/RCL {N} (TAM [CHS]) ; TI : N:)
Register O	Shortcut (TAM)	O		Easy access to register O (R209) : General purpose register O (STO/RCL {O} (TAM [EE]) ; TI : O:)
Register P	Shortcut (TAM)	P		Easy access to register P (R210) : Distribution parameter : probability (STO/RCL P ([REG.1] F3) ; TI : P:)
Register Q	Shortcut (TAM)	Q		Easy access to register Q (R211) : Distribution parameter : shape/3rd integer (STO/RCL Q ([REG.1] F4) ; TI : Q:)
Register R	Shortcut (TAM)	R		Easy access to register R (R212) : Distribution parameter : rate (STO/RCL R ([REG.1] F5) ; TI : R:)

FullName	Type	Label	Catalog	Description
Register S	Shortcut (TAM)	S		Easy access to register S (R213) : Distribution parameter : scale (STO/RCL S ([REG.1]F6) ; TI : S:)
Register T	Shortcut (TAM)	T		Easy access to register T (R220) : Stack level 4 (STO/RCL T (TAM) ; TI : T:)
Register T	Variable (register)	T		Register T (reserved variable)
Register U	Shortcut (TAM)	U		Easy access to register U (R214) : General purpose register U (STO/RCL U ([REG.2]F6) ; TI : U:)
Register V	Shortcut (TAM)	V		Easy access to register V (R215) : General purpose register V (STO/RCL V ([REG.2]fF1) ; TI : V:)
Register W	Shortcut (TAM)	W		Easy access to register W (R216) : General purpose register W (STO/RCL W ([REG.2]fF2) ; TI : W:)
Register X	Shortcut (TAM)	X		Easy access to register X (R217) : Stack level 1 (STO/RCL X (TAM) ; TI : X:)
Register X	Variable (register)	X		Register X (reserved variable)
Register Y	Shortcut (TAM)	Y		Easy access to register Y (R218) : Stack level 2 (STO/RCL Y (TAM) ; TI : Y:)
Register Y	Variable (register)	Y		Register Y (reserved variable)
Register Z	Shortcut (TAM)	Z		Easy access to register Z (R219) : Stack level 3 (STO/RCL Z (TAM) ; TI : Z:)
Register Z	Variable (register)	Z		Register Z (reserved variable)
Regression	MENU	REGR	REGR	Regression functions
Remainder	Function (dyadic)	RMD	RMD	Remainder of division of Y by X
Remark	Command	REM	REM	Comment line in program (maximum 160 characters) (REM ' ' α (ALPHA) menu)
ReMp261	User Program	ReMp261		ReMp261 test program, preloaded from testPgms.bin
Reset	Command	RESET	RESET	Reset the calculator ; fill registers R10-R34 with special values (Reset? ; Info : Startup from user saved backup file SAVFILES/C47.sav)
Reset f/g timers	Command	S.RESET	S.RESET	Safe reset, then toggle ON/OFF all accessibility related options: fg.FUL (*); g.2Tp [*]; SH.4s [ ] ; HOME.3 [*]
Reset fitting	Command	ResetF	ResetF	Reset BestF to default (1) (L.R. selected to 001)
Reset plot	Command	PLTRST	0	PLOTMENU only : reset all plot options and redraw graph (Options include boxes, crosses, lines, scales)
Reset stopwatch	Command	RESET		Reset the stopwatch ([←] (STOPW active))
Resets	MENU	RESETS	RESETS	Reset user menus to defaults ; delete user keyboard assignments
Return	Command (PEM)	RTN	RTN	Return from (sub)routine to calling routine
Return and skip	Command (PEM)	RTN+1	RTN+1	Return from (sub)routine and skip the instruction following the calling XEQ (intended use is to allow a program to function as a built in test function, which if true, returns using RTN and using RTN+1 otherwise)
RHO	Character	P		Character P (929)
rho lowercase	Character	p		Character p (961)
Ribbons	MENU	RIBBONS	RIBBONS	Selection of ribbons for MyMenu (Ribbons are predefined sets of buttons to assign to the primary (unshifted) row of MyMenu)
Right	Symbol	→		Move right (navigation) or arrow character (alpha selection menus) (8594)
Right	Arrow	→		Move right (navigation) or arrow character (alpha selection menus)
Right angle	Character	⊥		Character ⊥ (8735)
Right bit mask	Function (monadic)	MASKR	MASKR	Set right n bits to use as mask (MASKR ___ TamNonReg menu)
Right curly bracket	Character	}		Character } (125)
Right double quotation mark	Character	”		Character ” (8221)
Right French quotation mark	Character	»		Character » (187)
Right justify	Function (monadic)	RJ	RJ	Right justify (within word size) ; returns shift in X and result in Y
Right over left arrow	Character	↗		Character ↗ (alpha g[xẑy] ; Code : 8644)
Right parenthesis	Symbol	)		Close (right) parenthesis (41)

FullName	Type	Label	Catalog	Description
Right single quotation mark	Character	'		Character ' (8217)
Right square bracket	Character	]		Character ] (93)
Right tack	Character	┌		Character ┌ (8866)
Right-right	Arrow	->		Move all the way to the right (navigation) or arrow character (alpha selection menus)
Rightwards dashed arrow	Character	->		Character -> (8674)
Rightwards dashed arrow	Character	->		Character -> (8674)
Ring	Character	◦		Character ◦ (8728)
Roll down	Command	R↓	R↓	Roll down stack
Roll up	Command	R↑	R↑	Roll up stack
Root fit	Setting	RootF	RootF [ ]	Set BestF to include root curve fitting ; deselects OrthoF (Adds 16 to BestF ; TI : L.R. selected to <BestF>)
Rotate left	Function (monadic)	RL	RL	Rotate left with number of bits (trailing input) (RL __ TamNonReg menu)
Rotate left (1)	Function (monadic)	RL1	RL1	Shortcut to rotate left (1 bit)
Rotate left through Carry	Function (monadic)	RLC	RLC	Rotate left through Carry (RLC __ TamNonReg menu)
Rotate right	Function (monadic)	RR	RR	Rotate right with number of bits (trailing input) (RR __ TamNonReg menu)
Rotate right (1)	Function (monadic)	RR1	RR1	Shortcut to rotate right (1 bit)
Rotate right through Carry	Function (monadic)	RRC	RRC	Rotate right through Carry (RRC __ TamNonReg menu)
Round	Function (monadic)	ROUND	ROUND	Rounds to current display format (type real)
Round to decimal places	Function (monadic)	RDP	RDP	Rounds to n decimal places (RDP __ TamNonReg menu)
Round to integer	Function (monadic)	ROUNDI	ROUNDI	Rounds to next integer (max 1000 digits)
Round to significant digits	Function (monadic)	RSD	RSD	Rounds to number of significant digits, subject to rounding mode (RMODE) (RSD __ TamNonReg menu)
Rounding mode	Setting (pgm)	RMODE	RMODE <sub>0</sub>	Set floating point rounding mode ; used only for RSD ; also used when converting from the extended precision internal format to packed reals ; 0 : ½E = round half even ; 1 : ½↑ = round half up ; 2 : ½↓ = round half down ; 3 : ←0 = round away from zero ; 4 : →0 = round towards zero ; 5 : ⌈x⌉ = ceiling ; 6 : ⌊x⌋ = floor (RMODE _ TamNonReg menu)
Rounding mode?	Command	RMODE?	RMODE?	Get floating point rounding mode (Set by RMODE)
Row norm	Command	RNORM	RNORM	Row norm of matrix in X
Row sum	Command	RSUM	RSUM	Row sum of matrix in X
RPM to deg/s	Function (linked ; monadic)	RPM→deg/s		Convert rotation per minute to degree per second (deg/s:)
RPM to rad/s	Function (linked ; monadic)	RPM→rad/s		Convert rotation per minute to radian per second (rad/s:)
Run/Stop	Command	R/S		Run/Stop (Program)
Run/Stop	Command	STOP	STOP	Run/Stop (Program)
Rydberg constant	Constant (#42)	R <sub>∞</sub>		c.rydberg R <sub>∞</sub> = +1.097373156816 × 10 <sup>7</sup> (/mol)
S	Character	S		Character S (83)
S acute	Character	Ŝ		Character Ŝ (346)
s acute lowercase	Character	ś		Character ś (347)
S caron	Character	Š		Character Š (352)
s caron lowercase	Character	š		Character š (353)
S cedilla	Character	Ş		Character Ş (350)
s cedilla lowercase	Character	ş		Character ş (351)
s lowercase	Character	s		Character s (115)
s to ftn	Function (linked ; monadic)	s→ftn		Convert second to fortnight (ftn:)
s to year	Function (linked ; monadic)	s→year		Convert second to year (year:)
SAMPI	Character	ſ		Character ſ (992)
sampi lowercase	Character	ſ̣		Character ſ̣ (993)
Sample standard deviation	Command	s	s	Sample standard deviation (s <sub>x</sub> ; s <sub>y</sub> = (2 stack levels))
Satell	User Program	Satell		Satell test program, preloaded from testPgms.bin
Save full backup	Command (nonpgm)	SAVE	SAVE	Save full backup to file SAVFILES/C47.sav in FAT (Saved ; Directory : SAVFILES/)

FullName	Type	Label	Catalog	Description
Save full state file	Command (nonpgm)	SAVEST	SAVEST	Save full state file to file STATE/<state>.s47 in FAT (Saved ; DMCP File save dialog (STATE/))
Save program	Command	X.SAVE		Save XEQC program (X.SAVE __ TamNonReg menu)
Scatter plot	MENU (item)	SCATR	SCATR	Scatter plot of measurments
Scatter plot	MENU	SCATR		Scatter plot of measurments (CENTRL (re)starts menu SCATR)
Scattering factor for a lognormal population	Command	$\epsilon_p$	$\epsilon_p$	Scattering factor for a lognormal population ( $\epsilon_{px}$ ; $\epsilon_{py} = (2 \text{ stack levels})$ )
Scattering factor for a lognormal sample	Command	$\epsilon$	$\epsilon$	Scattering factor for a lognormal sample ( $\epsilon_x$ ; $\epsilon_y = (2 \text{ stack levels})$ )
Scattering factor of the geometric mean	Command	$\epsilon_m$	$\epsilon_m$	Scattering factor of the geometric mean ( $\epsilon_{mx}$ ; $\epsilon_{my} = (2 \text{ stack levels})$ )
Scientific display large reals	Setting (pgm)	SCIOVR	SCIOVR (•)	Change display to SCI for reals too large to display in full
Scientific notation	Setting (pgm)	SCI	SCI ( )	Set numeric display mode to SCientific notation with nn+1 digits (SCI __ TamNonReg menu)
Screenshot	Command	SNAP	SNAP	Save screenshot as bitmap ; if executed from the keyboard (g[EXIT]) also saves contents of stack or alpha buffer as text ; plays clicking sound ((DMCP) [<f>] + [EEX] ; Info : screenshot saved in file SCREENS/<YYYYMMDD-HHMMSS00>.BMP ; data appended to file DATA/<YYYYMMDD-HHMMSS00>REGS.TSV ; new datafile after timeout of 2 minutes)
Scroll down/Single step	Command (nonpgm)	⏴		Single Step
Scroll up/Backstep	Command (nonpgm)	⏵		Back Step
Second derivative	MENU	f''	f''	Second derivative menu
Seconds	Function (monadic)	SEC	SEC	Seconds (of time)
Section	Character	§		Character § (167)
Seed	Command	SEED	SEED	Set random seed (0..1] ; for values less than or equal to 0, the seed is derived from the internal clock
Select FP separator	MENU	FPART	FPART	Select fractional part separator
Select FP separator	MENU (item)	FPART_		Select fractional part separator ; button label showing current separator (.) ; symbol ø for no separator (CAT.MENUS FPART)
Select IP separator	MENU	IPART	IPART	Select integer part separator
Select IP separator	MENU (item)	IPART_		Select integer part separator ; button label showing current separator (.) ; symbol ø for no separator (CAT.MENUS IPART)
Select radix	MENU (item)	RADIX.		Select radix ; button label showing current radix (.) (CAT.MENUS RADIX)
Select radix	MENU	RADIX	RADIX	Select radix
Semicolon	Character	;		Character ; (alpha f[0] ; Code : 59)
Semi-major axis of the Earth	Constant (#45)	$S_{a\oplus}$		majax.earth $S_{a\oplus} = +6.378137 \times 10^6$ (m)
Semi-minor axis of the Earth	Constant (#46)	$S_{b\oplus}$		minax.earth $S_{b\oplus} = +6.3567523142 \times 10^6$ (m)
Serial I/O	Flag (system)	RUNIO		Serial input/output active (SBI depends on SBser)
Set ADM to D.MS	Setting (legacy)	D.MS	D.MS ( )	Set ADM to sexagesimal degrees mode (SBI depends on SBang)
Set ADM to DEG	Setting (pgm)	DEG	DEG (•)	Set ADM to degrees mode (SBI depends on SBang)
Set ADM to GRAD	Setting (pgm)	GRAD	GRAD ( )	Set ADM to gradians mode (SBI depends on SBang)
Set ADM to MULr	Setting (legacy)	MULr	MULr ( )	Set ADM multiple of pi radians mode (SBI depends on SBang)
Set ADM to RAD	Setting (pgm)	RAD	RAD ( )	Set ADM to radians mode (SBI depends on SBang)
Set all (models)	Command	AllF	AllF	Set BestF to include all curve fitting models (511) ; deselects OrthoF (L.R. selected to 511)
Set bit	Function (monadic)	SB	SB	Set bit n (SB __ TamNonReg menu)
Set current index	Command	STOIJ	STOIJ	Set current index (row I and column J) ( $[I_r, J_c] = [<row> <col>]$ )
Set D.MS tag or convert to D.MS	Function (monadic)	→D.MS	→D.MS	If untagged, set tag to D.MS ; if tagged, convert X to D.MS ; does not change ADM (° ' ")
Set date	Setting (pgm ; HW)	SETDAT	SETDAT	Set date ; hardware only
Set DEG tag or convert to DEG	Function (monadic)	→DEG	→DEG	If untagged, set tag to DEG ; if tagged, convert X to DEG ; does not change ADM (°)
Set flag	Command	SF	SF	Set flag (SF __ TamFlag menu)
Set GRAD tag or convert to GRAD	Function (monadic)	→GRAD	→GRAD	If untagged, set tag to GRAD ; if tagged, convert X to GRAD ; does not change ADM (°)
Set Julian-Gregorian (transition)	Function (monadic)	J/G	J/G	Set the day that Julian date changes over to Gregorian date (using date in X) (Input real number according to function x→DATE)



FullName	Type	Label	Catalog	Description
Set MUL $\pi$ tag or convert to MUL $\pi$	Function (monadic)	$\rightarrow$ MUL $\pi$	$\rightarrow$ MUL $\pi$	If untagged, set tag to MUL $\pi$ ; if tagged, convert X to MUL $\pi$ ; does not change ADM ("")
Set numeric (one digit)	Alpha-shift	di<g>it		Set numeric (one digit) (alpha g + <char>)
Set RAD tag or convert to RAD	Function (monadic)	$\rightarrow$ RAD	$\rightarrow$ RAD	If untagged, set tag to RAD ; if tagged, convert X to RAD ; does not change ADM ("")
Set significant digits	Setting (pgm)	SDIGS	SDIGS <sub>34</sub>	Set the number of significant digits (1 ... 34) for rounding after each operation ; sets tolerance of solvers and CONVG? ; value of 0 sets maximum precision (34) (SDIGS __ TamNonReg menu)
Set significant digits	Setting (pgm)	>SDIGS<		Old version of the setting (pgm) SDIGS that used stack input (May be found in older user programs)
Set time	Setting (pgm ; HW)	SETTIM	SETTIM	Set time ; hardware only
Setup	MENU	SETUP	SETUP	System (mode) settings with status indication and modification (Menu SETUP replaced by MODE in this layout)
Sexagesimal	Function (monadic)	.ms <sup>-1</sup>	.ms <sup>-1</sup>	Convert hh:mm:ss hours or dd°mm:ss" degrees to sexagesimal format number (untag) (hh/ddd.mmss: ; Info : puts sexagesimal NIM input of .ms back as decimal number on the stack)
Shade curve	Setting	[AREA		Shade the area underneath the graphical integral curve
Shift Digits Left	Function (monadic)	SDL	SDL	Shift digits to the left (SDL __ TamNonReg menu)
Shift Digits Right	Function (monadic)	SDR	SDR	Shift digits to the right (SDR __ TamNonReg menu)
Shift f	f-shift	f	f	Shift f (yellow) (SBI position depends on SBshfR)
Shift f/g	fg-shift	f/g	f/g	Single press : shift f (yellow) ; double press : shift g (blue) (SBI position depends on SBshfR)
Shift g	g-shift	g	g	Shift g (blue) (double [f/g] ; Info : SBI position depends on SBshfR)
Shift left	Function (monadic)	SL	SL	Shift bits right (SL __ TamNonReg menu)
Shift left (1)	Function (monadic)	SL1	SL1	Shortcut to shift left (1 bit)
Shift right	Function (monadic)	SR	SR	Shift bits right (SR __ TamNonReg menu)
Shift right (1)	Function (monadic)	SR1	SR1	Shortcut to shift right (1 bit)
Shift time-out 4s	Setting	SH.4s	SH.4s [ ]	Set shift to time out after 4 seconds
short cut to kg	Function (linked ; monadic)	short cwt $\rightarrow$ kg		Convert short hundredweight to kilogram (kg:)
Short integer	Function (monadic)	SINT	SINT	Convert to short integer
Short right arrow	Character	$\rightarrow$		Character $\rightarrow$ (8640)
short ton to kg	Function (linked ; monadic)	short ton $\rightarrow$ kg		Convert short ton to kilogram (kg:)
Shortint variables	MENU	S.INTS	S.INTS	Auto-generated catalog of variables of the specified type : shortint
Show	Command	SHOW	SHOW	Show item in maximum detail, favouring register data type (tag)
Show all prefixes	Setting	PFX.All	PFX.All [.]	Show all SI unit prefixes in UNIT (prefix) display mode: 10 <sup>-30</sup> to 10 <sup>30</sup> ; if OFF, the range is 10 <sup>-15</sup> to 10 <sup>15</sup>
Show angular mode symbol	Flag	SBang		Display symbol $\angle$ in status bar
Show battery capacity	Flag	SBbatV		Display battery voltage with battery length proportional to 2.054 V to 3.045 V (replacing the standard LOWBAT icon)
Show complex mode	Flag	SBcpx		Display status of RECT : Rectangular ; POLAR : Polar in status bar
Show complex result	Flag	SBcr		Display status of CPXRES : Complex results in status bar
Show date	Flag	SBdate		Display date in status bar
Show fraction and base mode	Flag	SBfrac		Display status of fractions, irrational fractions and number (base) in status bar
Show integer mode	Flag	SBint		Display status of WSIZE : Word size and complement/unsigned in status bar
Show matrix mode	Flag	SBmx		Display status of WRAP : Wrap (matrix edit) ; GROW : Grow (matrix edit) in status bar
Show oc mode	Flag	SBoc		Display status of OVERFL : Overflow ; CARRY : Carry in status bar
Show printer	Flag	SBprn		Display status of PRINTS : Printing in status bar
Show serial io	Flag	SBser		Display status of RUNIO : Serial I/O in status bar
Show shift on the right	Flag	SBshfR		Display <b>f</b> ; <b>g</b> on the right side of the status bar
Show stack size	Flag	SBss		Display status of SSIZE8 : Stack Size 8 in status bar
Show stopwatch	Flag	SBclk		Display status of RUNTIM : Stopwatch running in status bar
Show time	Flag	SBtime		Display time in status bar
Show tvn mode	Flag	SBtvn		Display status of End : TVM end payments ; Begin : TVM begin payments in status bar

FullName	Type	Label	Catalog	Description
Shuffle stack	Command	⌘	⌘	Shuffle stack: replace X Y Z T by selection of X Y Z T (⌘ ____ TamShuffle menu)
SIGMA	Character	Σ		Character Σ (alpha g[Σ+]; Code : 931)
Sigma-	Command	Σ-	Σ-	Remove data from the statistics matrix (STATS) (nnn data points)
sigma final lowercase	Character	ς		Character ς (962)
sigma lowercase	Character	σ		Character σ (963)
Sigma+	Command	Σ+	Σ+	Enter data into the statistics matrix (STATS) (nnn data point(s))
Sign	Function (monadic)	sign	sign	Sign is -1 for negative numbers, 0 for zero, +1 for positive numbers
Sign and mantissa	Setting (pgm)	SIGNMT	SIGNMT ( )	Set sign and mantissa mode for shortint (SBI depends on SBint)
Signed 16 bits	Setting (pgm)	S16		Shortcut to set word size to 16 bits signed
Signed 32 bits	Setting (pgm)	S32		Shortcut to set word size to 32 bits signed
Signed 6 bits	Setting (pgm)	S06		Shortcut to set word size to 6 bits signed
Signed 64 bits	Setting (pgm)	S64		Shortcut to set word size to 64 bits signed
Signed 8 bits	Setting (pgm)	S08		Shortcut to set word size to 8 bits signed
Significant digits notation	Setting (pgm)	SIG	SIG ( )	Set numeric display mode to SIGNificant notation with nn+1 digits; switching over to scientific or engineering notation when number of rounded trailing zeros exceeds group size (IPGRP) (SIG __ TamNonReg menu)
Significant digits?	Command	SDIGS?	SDIGS?	Get the number of significant digits (1 ... 34) for rounding after each operation (Set by SDIGS)
Silver ratio	Constant (#80)	δ <sub>s</sub>		r.silver δ <sub>s</sub> = +2.414213562373095048801688724209698 (δ <sub>s</sub> = 1 + √2)
Sinc	Function (monadic)	sinc	sinc	(Sine of X) / X (Integers and untagged reals are assumed to be radians)
Sinc pi	Function (monadic)	sincπ	sincπ	(Sine of π * X) / (π * X) (Integers and untagged reals are assumed to be radians)
Sine	Function (monadic)	SIN	SIN	Sine
Sine	Symbol	SIN		Sine (alpha g[SIN] ("SIN"))
Single high-reversed-9 quotation mark	Character	'		Character ' (8219)
Single low-9 quotation mark	Character	,		Character , (8218)
Skip	Command (PEM)	SKIP	SKIP	Skip n program steps (SKIP __ TamNonReg menu)
Slash	Character	/		Character / (47)
Slow mode	Flag	SLOW		Slow mode active to limit battery drain
SlvCnst	User Program	SlvCnst		SlvCnst test program, preloaded from testPgms.bin
SlvExtr	User Program	SlvExtr		SlvExtr test program, preloaded from testPgms.bin
Solve cubic	Command	SLVC	SLVC	Solve the cubic equation : a3 * x <sup>3</sup> + a2 * x <sup>2</sup> + a1 * x + a0 = 0 (a3 ENTER a2 ENTER a1 ENTER a0 ; roots 1, 2, 3 in Z, Y, X)
Solve quadratic	Command	SLVQ	SLVQ	Solve the quadratic equation : a * x <sup>2</sup> + b * x + c = 0 (a ENTER b ENTER c ; roots 1, 2 in Y, X)
Solver (EQN variables)	MENU	∫ f	∫ f	Integrator (variables) menu for the algebraic expression entered in NEW [EQN]; enter value and press <var <sub>n</sub> > button(s) to set variables, then press button for variable to be used by the integrator in this menu or in menu Tool] [EQN]; if only one variable exists it will be selected and starred automatically (TI : "Select Integrator Variable" is presented if Tool] is activated without a variable having been selected first)
Solver (EQN variables)	MENU	f Solve	f Solve	Solver (variables) menu for the algebraic expression ( = 0 ) entered in NEW [EQN]; enter value and press <var <sub>n</sub> > button(s) to initialise variable(s), then press button for variable to be solved for by the solver(s) in menu ToolS; if only one variable exists it will be selected and starred automatically (TI : "Select Solver Variable" is presented if ToolS is activated without a variable having been selected first)
Solver (RPN)	Command	SOLVE	SOLVE	Real solver for the RPN program (with MVAR variables) selected with menu TamLabel; presents menu for initial guesses and variable selection; interrupt by keypress; tolerance set by SDIGS; if MONIT is set, displays current evaluated top and bottom numbers and iteration counter; in PEM, RPN program must be specified by PGMSLV and variable with menu Tam; SOLVE will skip the next program step if no root is found (SOLVE __ TamLabel menu(uses Tam menu in PEM); TI : Accuracy ≈ ; <var1..n> = (n stack levels))

FullName	Type	Label	Catalog	Description
Solver tools	MENU	ToolS	ToolS	Solver tools ; parameter settings (Solver Result Codes ; Info : if MONIT is set, the solver displays current evaluated number and iteration counter)
Solving	Flag (system)	SOLVING		Solver is running
Sort registers	Command	R-SORT	R-SORT	Sort registers (sss.nn means sort registers from sss through sss + nn - 1)
Space	Character	' '		Character " " (32)
Space symbol	Character	␣		Character ␣ (9251)
Special key assignments	MENU	-> +NRM	-> +NRM	Special key assignments to [Σ+] key in normal mode
Special results	Setting	SPCRES	SPCRES [•]	Set to allow special results of calculations (infinity, not-a-number) ; an error will not occur for such events
Special results	Flag	SPCRES		Set to allow special results of calculations (infinity, not-a-number) ; an error will not occur for such events
Special results	Command	SPCRES1	SPCRES1	Allow special results of calculations (infinity, not-a-number) ; an error will not occur for such events (For programming purposes)
Special?	Function (monadic)	SPEC?	SPEC?	Test X is special (±∞ or NaN)
Specify program for ∫fd	Command	PGMINT	PGMINT	In PEM, specify the RPN program (with MVAR variables) to be used as the expression, store values in variable(s), then open menu ∫fd and select integrator and variable to be integrated for (PGMINT __ TamLabel menu)
Specify program for SOLVE	Command	PGMSLV	PGMSLV	In PEM, specify the RPN program (with MVAR variables) to be used as the expression ( = 0 ), store values in variable(s), then call SOLVE and specify variable to be solved for (PGMSLV __ TamLabel menu)
Speed	User Program	Speed		Speed test program, preloaded from testPgms.bin
Speed conversion	MENU	Speed:	Speed:	Convert between units of speed
Speed of light (vacuum)	Constant (#04)	c		lightspeed c = +2.99792458 × 10 <sup>8</sup> (m/s)
SPIRAL	User Program	SPIRAL		SPIRAL test program, preloaded from testPgms.bin
Square	Function (monadic)	x <sup>2</sup>	x <sup>2</sup>	Square of X
Square matrix?	Function (monadic)	M.SQR?	M.SQR?	Test matrix is square
Square root	Symbol	√		Square root (alpha g[√x] ; Code : 8730)
Square root	Function (legacy ; monadic)	SQRT	SQRT	Square Root (Superseded by √x : (keyboard [√x]))
Square root	Function (monadic)	√		Square root (8730)
Square root	Function (monadic)	√x	√x	Square root
SST	Character	∇		Character ∇ (8651)
SST	Character	∇		Character ∇ (8651)
Stack	MENU	STK	STK	Stack functions
Stack size 4	Setting (pgm)	SSIZE4	SSIZE4 ( )	Set stack size to 4 registers (SBI depends on SBss)
Stack Size 8	Setting (pgm)	SSIZE8	SSIZE8 (•)	Set stack size to 8 registers (SBI depends on SBss)
Stack Size 8	Flag	SSIZE8		Set for 8 stack registers, clear for 4 registers (SBI depends on SBss)
Stack size?	Command	SSIZE?	SSIZE?	Get stack size (Set by SSIZE4 or SSIZE8 ; Flag SSIZE8 is default 0N)
Stack to date	Function (triadic)	→DATE	→DATE	Convert stack values X, Y, Z to date according to DISP or CLK settings for date format (weekday)
Stack to Time	Function (monadic)	→TIME	→TIME	Convert to time
Standard atmospheric pressure	Constant (#37)	p <sub>0</sub>		press.atm p <sub>0</sub> = +1.01325 × 10 <sup>5</sup> (Pa)
Standard Earth acceleration	Constant (#17)	g <sub>⊕</sub>		acc.earth g <sub>⊕</sub> = +9.80665 (m/s <sup>2</sup> )
Standard error of the mean	Command	s <sub>m</sub>	s <sub>m</sub>	Standard error of the mean (s <sub>m<sub>x</sub></sub> ; s <sub>m<sub>y</sub></sub> = (2 stack levels))
Standard error of the weighted mean	Command	s <sub>m<sub>w</sub></sub>	s <sub>m<sub>w</sub></sub>	Standard error of the weighted mean (s <sub>m<sub>w</sub></sub> =)
Standard errors	Command	s(a)	s(a)	Standard errors of line fitted ; s(a <sub>0</sub> ) in X ; s(a <sub>1</sub> ) in Y (Works for ExpF ; LinF ; LogF ; OrthoF ; PowerF ; does not work for CauchF ; GaussF ; HypF ; ParabF ; RootF)
Standard normal (quantile)	Function (monadic)	Φ <sup>-1</sup>	Φ <sup>-1</sup>	Standard normal inverse cumulative distribution
Standard normal cdf (lower)	Function (monadic)	Φ <sub>⋄</sub>	Φ <sub>⋄</sub>	Standard normal cumulative distribution (lower tail)
Standard normal cdf (upper)	Function (monadic)	Φ <sub>⋄</sub>	Φ <sub>⋄</sub>	Standard normal cumulative distribution (upper tail)

FullName	Type	Label	Catalog	Description
Standard normal distribution	MENU	$\Phi$ :	$\Phi$ :	Standard normal probability distribution (continuous) (mean = 0 ; standard deviation = 1)
Standard normal pdf	Function (monadic)	$\varphi_p$	$\varphi_p$	Standard normal probability density function
Standard temperature	Constant (#50)	$T_0$		temp.stand $T_0 = +2.7315 \times 10^2$ (K)
Star (Wye) to Delta	Function (triadic)	$Y \rightarrow \Delta$	$Y \rightarrow \Delta$	Convert delta connected impedances X, Y, Z to star impedances X, Y, Z ( $\Delta \rightarrow Y$ )
Start/stop stopwatch	Command	R/S		Start/stop the stopwatch ([R/S] (STOPW active))
STAT000	User Program	STAT000		STAT000 test program, preloaded from testPgms.bin
STAT001	User Program	STAT001		STAT001 test program, preloaded from testPgms.bin
STAT002	User Program	STAT002		STAT002 test program, preloaded from testPgms.bin
STAT111	User Program	STAT111		STAT111 test program, preloaded from testPgms.bin
STAT113	User Program	STAT113		STAT113 test program, preloaded from testPgms.bin
STAT115	User Program	STAT115		STAT115 test program, preloaded from testPgms.bin
STATgr	User Program	STATgr		STATgr test program, preloaded from testPgms.bin
Statistics	MENU	STAT	STAT	Statistics functions
Statistics matrix	Variable (matrix)	STATS	STATS	Reserved matrix variable for statistics matrix (STATS) (Created from data entered using $\Sigma+$ ; cannot be deleted)
Stats range	Command	$\times$ RANGE	$\times$ RANGE	Range for both X and Y ; this is equal to MAX - MIN of the statistics matrix (STATS) ( $rg_x$ ; $rg_y$ = (2 stack levels))
Status	Command	STATUS	STATUS	Show status pages (memory, flags and settings)
STD_iota_TONOS.lo	Character	í		Character í (943)
STD_TIMER	Character	⌚		Character ⌚ (9201)
Stefan-Boltzmann constant	Constant (#72)	$\sigma_B$		c.stephbol $\sigma_B = +5.670374419184429453970996731889231 \times 10^{-9}$ ( $W/m^2K^4$ )
STO/RCL configuration	Command (TAM)	Config		Presented in TamStoRcl menu to quickly select STOCFG/RCLCFG (STO/RCL configuration) (STOCFG/RCLCFG __ TamStoRcl menu)
STO/RCL matrix element	Command (TAM)	...EL		Presented in TamStoRcl menu to quickly select STOEL/RCLEL (STO/RCL matrix element)
STO/RCL matrix index	Command (TAM)	...IJ		Presented in TamStoRcl menu to quickly select STOIJ/RCLIJ (STO/RCL matrix index)
STO/RCL maximum	Command (TAM)	Max		Presented in TamStoRcl menu to quickly select STO $\uparrow$ /RCL $\uparrow$ (STO/RCL maximum) (STO $\uparrow$ /RCL $\uparrow$ __ TamStoRcl menu)
STO/RCL minimum	Command (TAM)	Min		Presented in TamStoRcl menu to quickly select STO $\downarrow$ /RCL $\downarrow$ (STO/RCL minimum) (STO $\downarrow$ /RCL $\downarrow$ __ TamStoRcl menu)
STO/RCL stack	Command (TAM)	Stack		Presented in TamStoRcl menu to quickly select STOS/RCLS (STO/RCL stack) (STOS/RCLS __ STOS/RCLS)
stone to kg	Function (linked ; monadic)	stone $\rightarrow$ kg		Convert stone to kilogram (kg:)
Stopwatch	App	STOPW	STOPW	Stopwatch with running time and counter
Stopwatch	App (menu)	STOPW		Stopwatch with running time and counter ([0]... : Direct entry of digit 0... into CRA (Current Register Address), ...[9] : Direct entry of digit ...9 into CRA (Current Register Address), [ $\blacktriangle$ ] : Increment CRA (Current Register Address), [ $\blacktriangledown$ ] : Decrement CRA (Current Register Address), [RCL] : Recall and add register value to running time, g[EXIT] : Save screenshot as bitmap (SNAP), [EXIT] : Exit the stopwatch)
Stopwatch key [.]	Shortcut (stopwatch)	[.]		Keyboard shortcut for button "Store lap time in register"
Stopwatch key [+]	Shortcut (stopwatch)	[+]		Keyboard shortcut for button "Add lap time to STATS"
Stopwatch key [ $\leftarrow$ ]	Shortcut (stopwatch)	[ $\leftarrow$ ]		Keyboard shortcut for button "Reset stopwatch"
Stopwatch key [ENTER]	Shortcut (stopwatch)	[ENTER]		Keyboard shortcut for button "Store time in register"
Stopwatch key [R/S]	Shortcut (stopwatch)	[R/S]		Keyboard shortcut for button "Start/stop stopwatch"
Stopwatch key [ $\Sigma+$ ]	Shortcut (stopwatch)	[ $\Sigma+$ ]		Keyboard shortcut for button "Add time to STATS"
Stopwatch running	Flag (system)	RUNTIM		Set if the stopwatch is running (SBI not implemented yet ; Info : SBI depends on SBclk)
Store (register)	Command	STO	STO	Store value in register or variable ; can be followed by +, -, $\times$ , $\div$ for add into, subtract into, multiply into, divide into functions (STO __ TamStoRcl menu)
Store configuration	Command	STOCFG	STOCFG	Store configuration in register or variable (STOCFG __ TamStoRcl menu)

FullName	Type	Label	Catalog	Description
Store d <sub>1</sub> in M	Command	STO <sub>M</sub> d <sub>1</sub>		Store parameter d <sub>1</sub> in register M
Store d <sub>2</sub> in N	Command	STO <sub>N</sub> d <sub>2</sub>		Store parameter d <sub>2</sub> in register N
Store k in Q	Command	STO <sub>Q</sub> k		Store parameter k in register Q
Store K in Q	Command	STO <sub>Q</sub> K		Store parameter K in register Q
Store lap time in register	Command	LAP→R		Store lap time in register, increment CRA (Current Register Address) and reset lap timer ; subscript shows CRA ([.] (STOPW active))
Store maximum	Command	STO <sup>^</sup>	STO <sup>^</sup>	Store maximum of X and register or variable (STO <sup>^</sup> __ TamStoRcl menu)
Store minimum	Command	STO <sup>v</sup>	STO <sup>v</sup>	Store minimum of X and register or variable (STO <sup>v</sup> __ TamStoRcl menu)
Store n in N	Command	STO <sub>N</sub> n		Store parameter n in register N
Store N in M	Command	STO <sub>M</sub> N		Store parameter N in register M
Store p in P	Command	STO <sub>P</sub> p		Store parameter p in register P
Store s in S	Command	STO <sub>S</sub> s		Store parameter s in register S
Store stack	Command	STOS	STOS	Store entire stack in 4 or 8 registers (STOS __ TamStoRcl menu)
Store time in register	Command	TIM→R		Store time in register and increment CRA (Current Register Address) ; subscript shows CRA ([ENTER] (STOPW active))
Store triple I	Command	STO 3I	STO 3I	Copy X, Y, Z to R96, R97, R98
Store triple V	Command	STO 3V	STO 3V	Copy X, Y, Z to R93, R94, R95
Store triple Z	Command	STO 3Z	STO 3Z	Copy X, Y, Z to R90, R91, R92
Store X into current element	Command	STOEL	STOEL	Store X into current element of indexed matrix (M[ <sub>I</sub> , <sub>J</sub> ] = M[<row> <col>])
Store X into next element	Command	STOSEQ	STOSEQ	Store X into next element of indexed matrix (increase index and write element) (Increases row I and column J and accesses element)
Store X into vector element	Command	STOVEL	STOVEL	Store X into element n (TAM) of vector or matrix in Y (STOVEL __ Tam menu)
Store x <sub>n</sub> in M	Command	STO <sub>M</sub> x <sub>n</sub>		Store parameter x <sub>n</sub> in register M
Store y in S	Command	STO <sub>S</sub> y		Store parameter y in register S
Store λ in R	Command	STO <sub>R</sub> λ		Store parameter λ in register R
Store λ in S	Command	STO <sub>S</sub> λ		Store parameter λ in register S
Store μ in M	Command	STO <sub>M</sub> μ		Store parameter μ in register M
Store ν in M	Command	STO <sub>M</sub> ν		Store parameter ν in register M
Store ξ in Q	Command	STO <sub>Q</sub> ξ		Store parameter ξ in register Q
Store σ in S	Command	STO <sub>S</sub> σ		Store parameter σ in register S
String variables	MENU	STRINGS	STRINGS	Auto-generated catalog of variables of the specified type : string
String?	Function (monadic)	STRI?	STRI?	Test X is text string
Student's t (quantile)	Function (monadic)	t <sup>-1</sup> (p)	t <sup>-1</sup> (p)	Student's t inverse cumulative distribution
Student's t cdf (lower)	Function (monadic)	t <sub>λ</sub> (x)	t <sub>λ</sub> (x)	Student's t cumulative distribution (lower tail)
Student's t cdf (upper)	Function (monadic)	t <sub>λ</sub> (x)	t <sub>λ</sub> (x)	Student's t cumulative distribution (upper tail)
Student's t distribution	MENU	t:	t:	Student's t probability distribution (continuous) (RegM = ν = degrees of freedom)
Student's t pdf	Function (monadic)	t <sub>p</sub> (x)	t <sub>p</sub> (x)	Student's t probability density function
Subscript	Alpha-shift	↓ (α <sub>SUB</sub> )		Subscript (alpha g[▼])
Subscript -	Character	-		Character - (8331)
Subscript +	Character	+		Character + (8330)
Subscript 0	Character	0		Character 0 (8320)
Subscript 1	Character	1		Character 1 (8321)
Subscript 2	Character	2		Character 2 (8322)
Subscript 3	Character	3		Character 3 (8323)
Subscript 4	Character	4		Character 4 (8324)
Subscript 5	Character	5		Character 5 (8325)
Subscript 6	Character	6		Character 6 (8326)
Subscript 7	Character	7		Character 7 (8327)
Subscript 8	Character	8		Character 8 (8328)
Subscript 9	Character	9		Character 9 (8329)

FullName	Type	Label	Catalog	Description
Subscript A	Character	A		Character <sub>A</sub> (9424)
Subscript a lowercase	Character	a		Character <sub>a</sub> (9372)
Subscript alpha lowercase	Character	α		Character <sub>α</sub> (8854)
Subscript B	Character	B		Character <sub>B</sub> (9425)
Subscript b lowercase	Character	b		Character <sub>b</sub> (9373)
Subscript C	Character	C		Character <sub>C</sub> (9426)
Subscript c lowercase	Character	c		Character <sub>c</sub> (9374)
Subscript D	Character	D		Character <sub>D</sub> (9427)
Subscript d lowercase	Character	d		Character <sub>d</sub> (9375)
Subscript delta lowercase	Character	δ		Character <sub>δ</sub> (8855)
Subscript E	Character	E		Character <sub>E</sub> (8307)
Subscript e lowercase	Character	e		Character <sub>e</sub> (9376)
Subscript E small	Character	E		Character <sub>E</sub> (9428)
Subscript earth	Character	⊕		Character <sub>⊕</sub> (8853)
Subscript F	Character	F		Character <sub>F</sub> (9429)
Subscript f lowercase	Character	f		Character <sub>f</sub> (9377)
Subscript G	Character	G		Character <sub>G</sub> (9430)
Subscript g lowercase	Character	g		Character <sub>g</sub> (9378)
Subscript H	Character	H		Character <sub>H</sub> (9431)
Subscript h lowercase	Character	h		Character <sub>h</sub> (9379)
Subscript I	Character	I		Character <sub>I</sub> (9432)
Subscript i lowercase	Character	i		Character <sub>i</sub> (9380)
Subscript infinity	Character	∞		Character <sub>∞</sub> (8351)
Subscript J	Character	J		Character <sub>J</sub> (9433)
Subscript j lowercase	Character	j		Character <sub>j</sub> (9381)
Subscript K	Character	K		Character <sub>K</sub> (9434)
Subscript k lowercase	Character	k		Character <sub>k</sub> (9382)
Subscript L	Character	L		Character <sub>L</sub> (9435)
Subscript l lowercase	Character	l		Character <sub>l</sub> (9383)
Subscript M	Character	M		Character <sub>M</sub> (9436)
Subscript m lowercase	Character	m		Character <sub>m</sub> (9384)
Subscript mu lowercase	Character	μ		Character <sub>μ</sub> (8856)
Subscript N	Character	N		Character <sub>N</sub> (9437)
Subscript n lowercase	Character	n		Character <sub>n</sub> (9385)
Subscript O	Character	O		Character <sub>O</sub> (9438)
Subscript o lowercase	Character	o		Character <sub>o</sub> (9386)
Subscript P	Character	P		Character <sub>P</sub> (9439)
Subscript p lowercase	Character	p		Character <sub>p</sub> (9387)
Subscript pi	Character	π		Character <sub>π</sub> (subscript + alpha g[R+] ; Code : 11754)
Subscript Q	Character	Q		Character <sub>Q</sub> (9440)
Subscript q lowercase	Character	q		Character <sub>q</sub> (9388)
Subscript R	Character	R		Character <sub>R</sub> (9441)
Subscript r lowercase	Character	r		Character <sub>r</sub> (9389)
Subscript S	Character	S		Character <sub>S</sub> (9442)
Subscript s lowercase	Character	s		Character <sub>s</sub> (9390)
Subscript sun	Character	☉		Character <sub>☉</sub> (8858)
Subscript T	Character	T		Character <sub>T</sub> (9443)
Subscript t lowercase	Character	t		Character <sub>t</sub> (9391)
Subscript U	Character	U		Character <sub>U</sub> (9444)
Subscript u lowercase	Character	u		Character <sub>u</sub> (9392)

FullName	Type	Label	Catalog	Description
Subscript V	Character	v		Character v (9445)
Subscript v lowercase	Character	v		Character v (9393)
Subscript W	Character	w		Character w (9446)
Subscript w lowercase	Character	w		Character w (9394)
Subscript X	Character	x		Character x (9447)
Subscript x lowercase	Character	x		Character x (9395)
Subscript Y	Character	y		Character y (9448)
Subscript y lowercase	Character	y		Character y (9396)
Subscript Z	Character	z		Character z (9449)
Subscript z lowercase	Character	z		Character z (9397)
Subset Of	Character	c		Character c (8834)
Subtract	Function (dyadic)	-	-	Subtract X from Y
Subtract into	Function (monadic)	ST0-	ST0-	Subtract X from register or variable (ST0- __ TamStoRcl menu)
Sum (programmable)	Command	$\Sigma_n$	$\Sigma_n$	Real or complex sum using specified program ; interrupt by keypress ; if MONIT is set, displays current sum and iteration counter ( $\Sigma_n$ __ TamLabel menu ; <from> ENTER <to> ENTER <step>)
Summation of $\ln^2x$	Command	$\Sigma \ln^2x$	$\Sigma \ln^2x$	Summation of $\ln^2x$ using statistics matrix (STATS)
Summation of $\ln^2y$	Command	$\Sigma \ln^2y$	$\Sigma \ln^2y$	Summation of $\ln^2y$ using statistics matrix (STATS)
Summation of $\ln x$	Command	$\Sigma \ln x$	$\Sigma \ln x$	Summation of $\ln x$ using statistics matrix (STATS)
Summation of $\ln x \cdot \ln y$	Command	$\Sigma \ln x \cdot \ln y$	$\Sigma \ln x \cdot \ln y$	Summation of $\ln x \cdot \ln y$ using statistics matrix (STATS)
Summation of $\ln y$	Command	$\Sigma \ln y$	$\Sigma \ln y$	Summation of $\ln y$ using statistics matrix (STATS)
Summation of x	Command	$\Sigma x$	$\Sigma x$	Summation of x using statistics matrix (STATS)
Summation of $x^{-1}$	Command	$\Sigma x^{-1}$	$\Sigma x^{-1}$	Summation of $x^{-1}$ using statistics matrix (STATS)
Summation of $x \cdot \ln y$	Command	$\Sigma x \cdot \ln y$	$\Sigma x \cdot \ln y$	Summation of $x \cdot \ln y$ using statistics matrix (STATS)
Summation of $x^{-1} \cdot \ln y$	Command	$\Sigma x^{-1} \cdot \ln y$	$\Sigma x^{-1} \cdot \ln y$	Summation of $x^{-1} \cdot \ln y$ using statistics matrix (STATS)
Summation of $x^2$	Command	$\Sigma x^2$	$\Sigma x^2$	Summation of $x^2$ using statistics matrix (STATS)
Summation of $x^2 \cdot \ln y$	Command	$\Sigma x^2 \cdot \ln y$	$\Sigma x^2 \cdot \ln y$	Summation of $x^2 \cdot \ln y$ using statistics matrix (STATS)
Summation of $x^2 y$	Command	$\Sigma x^2 y$	$\Sigma x^2 y$	Summation of $x^2 y$ using statistics matrix (STATS)
Summation of $x^2 y^{-1}$	Command	$\Sigma x^2 y^{-1}$	$\Sigma x^2 y^{-1}$	Summation of $x^2 y^{-1}$ using statistics matrix (STATS)
Summation of $x^3$	Command	$\Sigma x^3$	$\Sigma x^3$	Summation of $x^3$ using statistics matrix (STATS)
Summation of $x^4$	Command	$\Sigma x^4$	$\Sigma x^4$	Summation of $x^4$ using statistics matrix (STATS)
Summation of $x^{-2}$	Command	$\Sigma x^{-2}$	$\Sigma x^{-2}$	Summation of $x^{-2}$ using statistics matrix (STATS)
Summation of xy	Command	$\Sigma xy$	$\Sigma xy$	Summation of xy using statistics matrix (STATS)
Summation of $xy^{-1}$	Command	$\Sigma xy^{-1}$	$\Sigma xy^{-1}$	Summation of $xy^{-1}$ using statistics matrix (STATS)
Summation of y	Command	$\Sigma y$	$\Sigma y$	Summation of y using statistics matrix (STATS)
Summation of $y^{-1}$	Command	$\Sigma y^{-1}$	$\Sigma y^{-1}$	Summation of $y^{-1}$ using statistics matrix (STATS)
Summation of $y \cdot \ln x$	Command	$\Sigma y \cdot \ln x$	$\Sigma y \cdot \ln x$	Summation of $y \cdot \ln x$ using statistics matrix (STATS)
Summation of $y^2$	Command	$\Sigma y^2$	$\Sigma y^2$	Summation of $y^2$ using statistics matrix (STATS)
Summation of $y^{-2}$	Command	$\Sigma y^{-2}$	$\Sigma y^{-2}$	Summation of $y^{-2}$ using statistics matrix (STATS)
Sun	Character	☉		Character ☉ (8857)
Superscript	Alpha-shift	↑ ( $\alpha^{SUP}$ )		Superscript (alpha g [▲])
Superscript -	Character	-		Character - (8555)
Superscript +	Character	+		Character + (8554)
Superscript 0	Character	0		Character 0 (8544)
Superscript 1	Character	1		Character 1 (8545)
Superscript -1	Character	-1		Character -1 (8306)
Superscript 2	Character	2		Character 2 (8546)
Superscript 3	Character	3		Character 3 (8547)
Superscript 4	Character	4		Character 4 (8548)
Superscript 5	Character	5		Character 5 (8549)

FullName	Type	Label	Catalog	Description
Superscript 6	Character	6		Character <sup>6</sup> (8550)
Superscript 7	Character	7		Character <sup>7</sup> (8551)
Superscript 8	Character	8		Character <sup>8</sup> (8552)
Superscript 9	Character	9		Character <sup>9</sup> (8553)
Superscript A	Character	A		Character <sup>A</sup> (9398)
Superscript a lowercase	Character	a		Character <sup>a</sup> (9346)
Superscript asterisk	Character	*		Character <sup>*</sup> (8335)
Superscript B	Character	B		Character <sup>B</sup> (9399)
Superscript b lowercase	Character	b		Character <sup>b</sup> (9347)
Superscript C	Character	C		Character <sup>C</sup> (9400)
Superscript c lowercase	Character	c		Character <sup>c</sup> (9348)
Superscript D	Character	D		Character <sup>D</sup> (9401)
Superscript d lowercase	Character	d		Character <sup>d</sup> (9349)
Superscript E	Character	E		Character <sup>E</sup> (9402)
Superscript e lowercase	Character	e		Character <sup>e</sup> (9350)
Superscript F	Character	F		Character <sup>F</sup> (9403)
Superscript f lowercase	Character	f		Character <sup>f</sup> (9351)
Superscript G	Character	G		Character <sup>G</sup> (9404)
Superscript g lowercase	Character	g		Character <sup>g</sup> (9352)
Superscript H	Character	H		Character <sup>H</sup> (9405)
Superscript h lowercase	Character	h		Character <sup>h</sup> (9353)
Superscript I	Character	I		Character <sup>I</sup> (9406)
Superscript i lowercase	Character	i		Character <sup>i</sup> (9354)
Superscript infinity	Character	∞		Character <sup>∞</sup> (8350)
Superscript J	Character	J		Character <sup>J</sup> (9407)
Superscript j lowercase	Character	j		Character <sup>j</sup> (9355)
Superscript K	Character	K		Character <sup>K</sup> (9408)
Superscript k lowercase	Character	k		Character <sup>k</sup> (9356)
Superscript L	Character	L		Character <sup>L</sup> (9409)
Superscript l lowercase	Character	l		Character <sup>l</sup> (9357)
Superscript M	Character	M		Character <sup>M</sup> (9410)
Superscript m lowercase	Character	m		Character <sup>m</sup> (9358)
Superscript N	Character	N		Character <sup>N</sup> (9411)
Superscript n lowercase	Character	n		Character <sup>n</sup> (9359)
Superscript O	Character	O		Character <sup>O</sup> (9412)
Superscript o lowercase	Character	o		Character <sup>o</sup> (9360)
Superscript P	Character	P		Character <sup>P</sup> (9413)
Superscript p lowercase	Character	p		Character <sup>p</sup> (9361)
Superscript pi	Character	π		Character <sup>π</sup> (superscript + alpha g [R+] ; Code : 11755)
Superscript Q	Character	Q		Character <sup>Q</sup> (9414)
Superscript q lowercase	Character	q		Character <sup>q</sup> (9362)
Superscript R	Character	R		Character <sup>R</sup> (9415)
Superscript r lowercase	Character	r		Character <sup>r</sup> (9363)
Superscript S	Character	S		Character <sup>S</sup> (9416)
Superscript s lowercase	Character	s		Character <sup>s</sup> (9364)
Superscript T	Character	T		Character <sup>T</sup> (9417)
Superscript t lowercase	Character	t		Character <sup>t</sup> (9365)
Superscript U	Character	U		Character <sup>U</sup> (9418)
Superscript u lowercase	Character	u		Character <sup>u</sup> (9366)
Superscript V	Character	V		Character <sup>V</sup> (9419)



FullName	Type	Label	Catalog	Description
Superscript v lowercase	Character	v		Character <sup>v</sup> (9367)
Superscript W	Character	W		Character <sup>W</sup> (9420)
Superscript w lowercase	Character	w		Character <sup>w</sup> (9368)
Superscript X	Character	X		Character <sup>X</sup> (9421)
Superscript x lowercase	Character	x		Character <sup>x</sup> (9369)
Superscript Y	Character	Y		Character <sup>Y</sup> (9422)
Superscript y lowercase	Character	y		Character <sup>y</sup> (9370)
Superscript Z	Character	Z		Character <sup>Z</sup> (9423)
Superscript z lowercase	Character	z		Character <sup>z</sup> (9371)
Surface integral	Character	∫		Character ∫ (8751)
survey ft <sub>US</sub> to m	Function (linked ; monadic)	survey ft <sub>US</sub> →m		Convert US survey foot to meter (m)
Swap registers	Command	R-SWAP	R-SWAP	Swap registers (sss.nnddd means swap registers from sss through sss + nn - 1 with registers ddd through ddd + nn - 1)
Swap rows	Command	R↔R	M.R↔R	Swap rows X and Y of indexed matrix
Swap T	Command	t↔	t↔	Swap T and register (t↔ __ Tam menu)
Swap X	Command	x↔	x↔	Swap X and register (x↔ __ Tam menu)
Swap X	Command	X.SWAP	X.SWAP	Swap contents of X register with input of alpha or equation editor
Swap X and Y	Command	x↔y	x↔y	Swap register X and register Y
Swap Y	Command	y↔	y↔	Swap Y and register (y↔ __ Tam menu)
Swap Z	Command	z↔	z↔	Swap Z and register (z↔ __ Tam menu)
Switch on	Command	ON		Switch on calculator (switched OFF + [EXIT])
S <sub>xy</sub>	Command	S <sub>xy</sub>	S <sub>xy</sub>	Sample covariance
System	Command (HW)	SYSTEM	SYSTEM	Exit calculator to enter DMCP (causing reset) ; write autobackup to SAVFILES/C47auto.sav ; hardware only (Exit to system? ; Info : on restart, restore from SAVFILES/C47auto.sav if available)
System flags	MENU (TAM ; ASM)	SYS.FL	SYS.FL	Presented in TAM menus for commands accessing system flags (Type characters 1-2 to search ; If SYS.FL is accessed from CAT.MENUS, soft buttons return flag status, otherwise toggle flag status)
SZ	Character	β		Character β (223)
sz lowercase	Character	β		Character β (223)
T	Character	T		Character T (84)
t apostrophe lowercase	Character	ƒ		Character ƒ (357)
T caron	Character	ř		Character ř (356)
T cedilla	Character	Ț		Character Ț (354)
t cedilla lowercase	Character	ț		Character ț (355)
t lowercase	Character	t		Character t (116)
TAM (flag)	MENU (TAM)	FLG		Presented in TamFlag menu to quickly select lettered flags (Shortcut {A} indicates character key A (in TAM), etcetera)
TAM (register)	MENU (TAM)	REG		Presented in Tam and TamStoRcl menu to quickly select lettered registers (Shortcut {A} indicates character key A (in TAM), etcetera)
TAM mode (alpha)	MENU (TAM)	TamAlpha		Transient Alpha Mode is activated for trailing input (alpha) (double [XEQ] ; Shortcut (TAM) : TAM [XEQ] ; Info : Activated from TAM menus by AIM (α))
TAM mode (compare)	MENU (TAM)	TamCmp		Transient Alpha Mode is activated for trailing input (compare) (AIM (α) activates TamAlpha)
TAM mode (flag)	MENU (TAM)	TamFlag		Transient Alpha Mode is activated for trailing input (flag)
TAM mode (general)	MENU (TAM)	Tam		Transient Alpha Mode is activated for trailing input (general) (AIM (α) activates TamAlpha)
TAM mode (indirect)	MENU (TAM)	TamNonRegInd		Transient Alpha Mode is activated for trailing input (indirect) (Activated from TAM menus by indirection (→))
TAM mode (label)	MENU (TAM)	TamLabel		Transient Alpha Mode is activated for trailing input (label) (AIM (α) activates TamAlpha)
TAM mode (shuffle)	MENU (TAM)	TamShuffle		Transient Alpha Mode is activated for trailing input (shuffle)

FullName	Type	Label	Catalog	Description
TAM mode (store/recall)	MENU (TAM)	TamStoRcl		Transient Alpha Mode is activated for trailing input (store/recall) (AIM (α) activates TamAlpha)
TAM mode (value)	MENU (TAM)	TamNonReg		Transient Alpha Mode is activated for trailing input (value) (Indirection (→) activates TamNonRegInd)
TAM mode KEY	MENU (TAM)	TamKey		Transient Alpha Mode is activated for trailing input (KEYG ; KEYX)
Tangent	Function (monadic)	TAN	TAN	Tangent
Tangent	Symbol	TAN		Tangent (alpha g [TAN] ("TAN"))
TAU	Character	T		Character T (932)
Tau	Constant (#82)	τ		c.circle τ = +6.283185307179586476925286766559006 (Moving sofa problem)
tau lowercase	Character	τ		Character τ (964)
Tenths of seconds	Setting	0.1s		Show running time with tenths of seconds
Tera	Command (nonpgm)	·T		Factor 10 <sup>12</sup>
Tera binary	Command (nonpgm)	·Ti		Factor 2 <sup>40</sup>
Testing	MENU	TEST	TEST	Testing functions
There does not exist	Character	∄		Character ∄ (8708)
There exists	Character	∃		Character ∃ (8707)
THETA	Character	Θ		Character Θ (920)
theta lowercase	Character	ϑ		Character ϑ (952)
Tick	Character	'		Character ' (700)
Ticks	Command	TICKS	TICKS	Number of ticks counted since calculator was turned on (one tick is 10 ms)
Tilde	Character	~		Character ~ (126)
Time	Command	TIME	TIME	Current time
Time display 24h	Flag	TDM24		Set for 24h time display, clear for 12h time (HH24:MM ; HH:MMam/pm)
Time display format	Setting	TDISP	TDISP <sub>0</sub>	Set time display format: 0 = full ; 1,2 = hours/minutes ; 3 = hours/minutes/seconds ; 4,5,6 = 1,2,3 decimal places for the seconds (TDISP _ TamNonReg menu)
Time to stack	Function (monadic)	TIME→	TIME→	Convert time to hours (24h), minutes, seconds in stack
Time value of money	MENU	TVM	TVM	Time value of money functions (TVM variable values are displayed as subscripts (F1-F5) ; symbol ∞ is used for very small values ; symbol ∞ for very large values)
Time variables	MENU	TIMES	TIMES	Auto-generated catalog of variables of the specified type : time
To integer	Function (cyclic ; monadic)	→I	→I	Convert to long integer/short integer (cyclic, max 1000 digits) (Shortint indicated by subscript <sub>10</sub> ; can show TI: Overfl>nbits (n=1..64))
To polar	Function (monadic ; dyadic)	→P	→POLAR	Transform rectangular to polar coordinates (stack conventions according to flag RP <sub>HP</sub> or ADM tag) ; transform complex number to polar notation (monadic) and set POLAR tag (r = ; ϑ = (2 stack levels))
To rectangular	Function (monadic ; dyadic)	→R	→RECT	Transform polar to rectangular coordinates (stack conventions according to flag RP <sub>HP</sub> ) ; transform complex number to rectangular notation (monadic) and set RECT tag (x : Re = ; y : Im = (2 stack levels))
ton to kg	Function (linked ; monadic)	ton→kg		Convert ton to kilogram (kg)
Tone	Command	TONE	TONE	Tone (range 0-11) (TONE _ TamNonReg menu)
Top?	Command	TOP?	TOP?	Test program pointer is in top routine (as opposed to in subroutine)
torr to Pa	Function (linked ; monadic)	torr→Pa		Convert torr to Pascal (Pa)
tr.oz to g	Function (linked ; monadic)	tr.oz→g		Convert troy ounce to gram (g)
Tracing	Flag	TRACE		Printing in trace mode
Transpose matrix	Function (monadic)	[M] <sup>T</sup>	[M] <sup>T</sup>	Transpose matrix in X
Transpose matrix	Function (dyadic)	TRANS	TRANS	Transpose matrix in X (Superseded by [M] <sup>T</sup> )
Transposed	Character	T		Character T (8868)
Trigonometry	MENU	TRG_D47	TRG_D47	Trigonometry functions (other layouts) (Menu TRG_D47 replaced by TRG in this layout ; Ref : DMS-HMS)
Trigonometry	MENU (47)	TRG	TRG	Trigonometry and hyperbolic functions (DMS-HMS)

FullName	Type	Label	Catalog	Description
Triple I = V / Z	Command	V÷Z	3V÷3Z	X = R93 / R90 ; Y = R94 / R91 ; Z = R95 / R92
Triple V = I x Z	Command	I×Z	3I×3Z	X = R96 × R90 ; Y = R97 × R91 ; Z = R98 × R92
Triple Z = V / I	Command	V÷I	3V÷3I	X = R93 / R96 ; Y = R94 / R97 ; Z = R95 / R98
TVM begin payments	Setting (pgm)	Begin	BeginP ( )	Payments at the beginning of each period (time value of money) (SBI depends on SBTvm)
TVM end payments	Setting (pgm)	End	ENDP (•)	Payments at the end of each period (time value of money) (SBI depends on SBTvm)
TVM end payments	Flag (system)	ENDPMT		Set for payments at the end of each period, clear for beginning (time value of money) (SBI depends on SBTvm)
U	Character	U		Character U (85)
U acute	Character	Ú		Character Ú (218)
u acute lowercase	Character	ú		Character ú (250)
U breve	Character	Û		Character Û (364)
u breve lowercase	Character	û		Character û (365)
U circumflex	Character	Û		Character Û (219)
u circumflex lowercase	Character	û		Character û (251)
U diaeresis	Character	Ü		Character Ü (220)
u diaeresis lowercase	Character	ü		Character ü (252)
U grave	Character	Û		Character Û (217)
u grave lowercase	Character	û		Character û (249)
u lowercase	Character	u		Character u (117)
U macron	Character	Ū		Character Ū (362)
u macron lowercase	Character	ū		Character ū (363)
U ogonek	Character	Ų		Character Ų (370)
u ogonek lowercase	Character	ų		Character ų (371)
U ring	Character	Ū		Character Ū (366)
u ring lowercase	Character	ū		Character ū (367)
U tilde	Character	Ū		Character Ū (360)
u tilde lowercase	Character	ü		Character ü (361)
UK	Character	UK		Character UK (9261)
UK formatting	Setting (pgm)	UK	SETUK	Set to UK regional formats (date, time, calendar, number formatting) (First Gregorian day set: 14.09.1752)
Underscore	Character	_		Character _ (95)
Undo	Command (nonpgm)	↶	UNDO	Restore complete stack, LASTx, STATS and system flags
Undo	Character	↶		Character ↶ (8653)
Union	Character	∪		Character ∪ (8746)
Unit (prefix) notation	Setting (pgm)	UNIT	UNIT ( )	Set numeric display mode to UNIT nn+1 digits ; showing all or limited set of SI-prefixes depending on setting PFX.All ; if setting 1024 <sup>n</sup> is set, binary prefixes are shown ; prefixes can be selected directly from menu Prefix (UNIT __ TamNonReg menu)
Unit in the last place?	Command	ULP?	ULP?	Get minimum difference to next or previous machine representable real, as power of ten
Unit vector	Command	UNITV	UNITV	Unit vector for complex number or matrix
Unsigned	Setting (pgm)	UNSIGN	UNSIGN ( )	Set unsigned mode for shortint (SBI depends on SBint)
Unsigned 16 bits	Setting (pgm)	U16		Shortcut to set word size to 16 bits unsigned
Unsigned 32 bits	Setting (pgm)	U32		Shortcut to set word size to 32 bits unsigned
Unsigned 6 bits	Setting (pgm)	U06		Shortcut to set word size to 6 bits unsigned
Unsigned 64 bits	Setting (pgm)	U64		Shortcut to set word size to 64 bits unsigned
Unsigned 8 bits	Setting (pgm)	U08		Shortcut to set word size to 8 bits unsigned
Up	Arrow	↑		Move up (navigation) or arrow character (alpha selection menus)
Up	Symbol	↑		Move up (navigation) or arrow character (alpha selection menus) (8593)
Up-down arrow	Character	↕		Character ↕ (8597)
Upper limit	Variable (real)	↑Lim	↑LIM	Upper limit for solvers and integrator (reserved real variable) ; displays as ↑L in menu Tools or Tool↓ when value ≠ 0 ; set interactively and by realSlv <sup>x</sup> , cpXSlv <sup>x</sup> or ∫ <sup>x</sup> (↑LIM :)

FullName	Type	Label	Catalog	Description
Upper limit of drawing interval	Variable (real)	↑X	↑X	Upper limit of drawing interval (reserved real variable) ; set interactively and by Draw <sup>x</sup>
Upper quantile	Command	x <sub>Q3</sub>	x <sub>Q3</sub>	Upper quantile for both X and Y (Q <sub>3</sub> x ; Q <sub>3</sub> y = (2 stack levels))
UPSILON	Character	Υ		Character Υ (933)
UPSILON dialytika	Character	Ÿ		Character Ÿ (939)
upsilon dialytika lowercase	Character	ÿ		Character ÿ (971)
upsilon dialytika tonos lowercase	Character	Ϸ		Character Ϸ (944)
upsilon lowercase	Character	υ		Character υ (965)
upsilon tonos lowercase	Character	ύ		Character ύ (973)
Upwards dashed arrow	Character	↕		Character ↕ (8673)
Upwards dashed arrow	Character	⤴		Character ⤴ (8673)
US	Character	us		Character us (9262)
USA formatting	Setting (pgm)	USA	SETUSA	Set to USA regional formats (date, time, calendar, number formatting) (First Gregorian day set: 9/14/1752)
USB Power	Flag (system)	USB		Calculator is connected to USB power
USB symbol	Character	ψ		Character ψ (9268)
User mode	Character	U		Character U (9260)
USER mode	Setting	USER		Set USER mode (<none> (KEY.LP ON))
USER mode	Flag	USER		Set for USER mode
V	Character	V		Character V (86)
v lowercase	Character	v		Character v (118)
V47 layout	Layout (SIM)	V47		V47 keyboard layout ; this layout works in USER mode on the simulator (V47: Exp Vintage 2 shifts TopR ->x/ L)
Vacuum electric permittivity	Constant (#60)	ε <sub>0</sub>		epermt.vac ε <sub>0</sub> = +8.8541878128 × 10 <sup>-12</sup> (As/Vm)
Vacuum magnetic permeability	Constant (#64)	μ <sub>0</sub>		mpermb.vac μ <sub>0</sub> = +1.25663706212 × 10 <sup>-6</sup> (Vs/Am)
VAR	MENU (TAM ; ASM)	VAR	VAR	Presented in TAM menus for commands accessing variables (CAT.VARS.* menu ; Type characters 1-2 to search)
Variable ...	Variable (internal)	<var...>		Next variable(s) if any, of the expression entered in NEW [EQN] (Starred variable <var...>* is selected for subsequent operation)
Variable ...	Variable (internal)	<var...>		Next variable(s) if any, as created by MVAR (in RPN program) (Starred variable <var...>* is selected for subsequent operation)
Variable 1	Variable (internal)	<var <sub>1</sub> >		Variable 1 of the expression entered in NEW [EQN] (Starred variable <var <sub>1</sub> >* is selected for subsequent operation)
Variable 1	Variable (internal)	<var <sub>1</sub> >		Variable 1 as created by MVAR (in RPN program) (Starred variable <var <sub>1</sub> >* is selected for subsequent operation)
Variable 2	Variable (internal)	<var <sub>2</sub> >*		Variable 2 of the expression entered in NEW [EQN] (starred) (Starred variable <var <sub>2</sub> >* is selected for subsequent operation)
Variable 2	Variable (internal)	<var <sub>2</sub> >*		Variable 2 as created by MVAR (in RPN program) (starred) (Starred variable <var <sub>2</sub> >* is selected for subsequent operation)
Variable menu	MENU (item ; PEM)	VarMNU	VARMNU	Create variable menu (VARMNU __ TamLabel menu)
Variable Menu Displayed	Flag (system)	VMDISP		Variable menu is displayed
Variable x	Symbol	x		Variable x
Variables	MENU	VARS	VARS	Auto-generated variable catalogs
Vector angle	Function (dyadic)	V∠	V∠	Angle between two vectors [a1 b1] and [a2 b2] or [a1 b1 c1] and [a2 b2 c2]
Version?	Command (nonpgm)	VERS?	VERS?	Show firmware version (Firmware version)
View	Command	VIEW	VIEW	View register or variable (VIEW __ Tam menu)
Volume	Setting (pgm ; HW)	VOL	VOL <sub>11</sub>	Set audio volume (range 0-11) ; hardware only (VOL _ TamNonReg menu)
Volume conversion	MENU	Volume:	Volume:	Convert between units of volume
Volume down	Setting (HW)	VOL↓	VOL↓ <sub>11</sub>	Volume down (range 0-11) ; hardware only
Volume of ideal gas	Constant (#53)	V <sub>m</sub>		volume.gas V <sub>m</sub> = +2.241396954501413773501110288675056 × 10 <sup>-2</sup> (m <sup>3</sup> /mol)

FullName	Type	Label	Catalog	Description
Volume up	Setting (HW)	VOL↑	VOL↑ <sub>11</sub>	Volume up (range 0-11) ; hardware only
Volume?	Command (HW)	VOL?	VOL?	Get audio volume (range 0-11) ; hardware only (Set by VOL)
Von Klitzing constant	Constant (#40)	R <sub>K</sub>		c.klitzing R <sub>K</sub> = +2.581280745930450666004551670608744 × 10 <sup>4</sup> (Ω)
W	Character	W		Character W (87)
W <sup>-1</sup>	Function (monadic)	W <sup>-1</sup>	W <sup>-1</sup>	Inverse of W <sub>p</sub> (≧ -1)
W circumflex	Character	Ŵ		Character Ŵ (372)
w circumflex lowercase	Character	ŵ		Character ŵ (373)
w lowercase	Character	w		Character w (119)
W to hp <sub>UK</sub>	Function (linked ; monadic)	W→hp <sub>UK</sub>		Convert Watt to UK horsepower (hp <sub>UK</sub> ;) )
W to hp <sub>E</sub>	Function (linked ; monadic)	W→hp <sub>E</sub>		Convert Watt to electrical horsepower (hp <sub>E</sub> ;) )
W to hp <sub>M</sub>	Function (linked ; monadic)	W→hp <sub>M</sub>		Convert Watt to metric horsepower (hp <sub>M</sub> ;) )
Watch	Character	⌚		Character ⌚ (8986)
Weekday	Function (monadic)	WDAY	WDAY	Weekday (of date) where 1 = Monday ... 7 = Sunday and show weekday in TI (Weekday)
Weibull (quantile)	Function (monadic)	Weibl <sup>-1</sup>	Weibl <sup>-1</sup>	Weibull inverse cumulative distribution
Weibull cdf (lower)	Function (monadic)	Weibl <sub>⋈</sub>	Weibl <sub>⋈</sub>	Weibull cumulative distribution (lower tail)
Weibull cdf (upper)	Function (monadic)	Weibl <sup>⋈</sup>	Weibl <sup>⋈</sup>	Weibull cumulative distribution (upper tail)
Weibull distribution	MENU	Weibl:	Weibl:	Weibull probability distribution (continuous) (RegQ = k = shape ; RegS = λ = scale)
Weibull pdf	Function (monadic)	Weibl <sub>p</sub>	Weibl <sub>p</sub>	Weibull probability density function
Weighted mean	Command	$\bar{x}_w$	$\bar{x}_w$	Weighted means of x with weight y ( $\bar{x}_w =$ )
Weighted population standard deviation	Command	$\sigma_w$	$\sigma_w$	Weighted population standard deviation ( $\sigma_w =$ )
Weighted population standard deviation	Command	s <sub>w</sub>	s <sub>w</sub>	Weighted population standard deviation (s <sub>w</sub> = )
Wh to J	Function (linked ; monadic)	Wh→J		Convert Watt-hour to Joule (J:)
Who? (team)	Command (nonpgm)	WHO?	WHO?	Show calculator development team names (Team names)
Wide comma	Character	,		Character , (10120)
Wide dot	Character	•		Character • (8901)
Wide period	Character	.		Character . (10121)
W <sub>m</sub>	Function (monadic)	W <sub>m</sub>	W <sub>m</sub>	Lambert's W function (negative branch ; m = minus)
Word size	Setting (pgm)	WSIZE	WSIZE <sub>64</sub>	Set word size in bits, for shortint (WSIZE __ TamNonReg menu ; Info : SBI depends on SBint)
Word size 16 bits	Setting (pgm)	16-BIT	16-BIT ( )	Set word size to 16 bits for shortint
Word size 32 bits	Setting (pgm)	32-BIT	32-BIT ( )	Set word size to 32 bits for shortint
Word size 64 bits	Setting (pgm)	64-BIT	64-BIT (•)	Set word size to 64 bits for shortint
Word size 8 bits	Setting (pgm)	8-BIT	8-BIT ( )	Set word size to 8 bits for shortint
Word size?	Command	WSIZE?	WSIZE?	Get word size in bits, for shortint (Bits = ; Info : set by WSIZE)
Word swap	Function (monadic)	W.SWP	W.SWP	Swap words
W <sub>p</sub>	Function (monadic)	W <sub>p</sub>	W <sub>p</sub>	Lambert's W function (principal branch)
Wrap (matrix edit)	Setting	WRAP	M.WRAP (•)	Matrix edit in wrapping mode (SBI depends on SBmx)
Wrap edge	Flag (system)	WRPEDG		Set if a matrix command causes the index to be wrapped from one edge to the opposite edge ; cleared otherwise
Wrap end	Flag (system)	WRPEND		Set if a matrix command causes the index to be wrapped from the first element to the last or vice versa ; cleared otherwise
Write program	Command	WRITEP	WRITEP	Write program to file PROGRAMS/<program>.p47 in FAT (WRITEP _ TamLabel ; DMCP : File save dialog (PROGRAMS/) ; TI : Saved)
X	Character	X		Character X (88)
X	User Program	X		X test program, preloaded from testPgms.bin
X approximates?	Function (monadic)	x≈ ?	x≈ ?	X approximates? (rounded values are equal) (x≈ ? __ TamCmp menu)
X Balanced	Function (monadic)	X→BAL	X→BAL	Create balanced 3 phase quantities by pushing onto stack X × a, and then X × a × a
x circumflex lowercase	Character	ẋ		Character ẋ (889)
X equals +0?	Function (monadic)	x=+0 ?	x=+0 ?	X equals +0? (shortint 1COMPL or SIGNMT) (x=+0 ? __ TamCmp menu)
X equals -0?	Function (monadic)	x=-0 ?	x=-0 ?	X equals -0? (shortint 1COMPL or SIGNMT) (x=-0 ? __ TamCmp menu)
X equals?	Function (monadic)	x= ?	x= ?	X equals? (x= ? __ TamCmp menu)

FullName	Type	Label	Catalog	Description
X greater or equal?	Function (monadic)	$x \geq ?$	$x \geq ?$	X greater or equal? ( $x \geq ?$ ___ TamCmp menu)
X greater?	Function (monadic)	$x > ?$	$x > ?$	X greater? ( $x > ?$ ___ TamCmp menu)
x hat	Command	$\hat{x}$	$\hat{x}$	Estimation of x
X less or equal?	Function (monadic)	$x \leq ?$	$x \leq ?$	X less or equal? ( $x \leq ?$ ___ TamCmp menu)
X less?	Function (monadic)	$x < ?$	$x < ?$	X less? ( $x < ?$ ___ TamCmp menu)
x lowercase	Character	x		Character x (120)
x macron lowercase	Character	$\bar{x}$		Character $\bar{x}$ (888)
X not equal?	Function (monadic)	$x \neq ?$	$x \neq ?$	X not equal? ( $x \neq ?$ ___ TamCmp menu)
X to alpha	Function (monadic)	$x \rightarrow \alpha$	$x \rightarrow \alpha$	Convert character code in X to alpha character (code value pushed to Y)
X to date	Function (monadic)	$x \rightarrow \text{DATE}$	$x \rightarrow \text{DATE}$	Convert date input number YYYY-MM-DD or DD.MM.YYYY or MM/DD/YYYY to date according to DISP or CLK format settings (Weekday)
x under root	Character	$\bar{x}$		Character $\bar{x}$ (895)
XEQ XEQM01	Command	XEQM01	XEQM01	Execute XEQM01 (HELP!!)
XEQ XEQM02	Command	XEQM02	XEQM02	Execute XEQM02 (BATPLT)
XEQ XEQM03	Command	XEQM03	XEQM03	Execute XEQM03 (MP2203)
XEQ XEQM04	Command	XEQM04	XEQM04	Execute XEQM04 (MP2281)
XEQ XEQM05	Command	XEQM05	XEQM05	Execute XEQM05 (MP3217)
XEQ XEQM06	Command	XEQM06	XEQM06	Execute XEQM06 (CUBES)
XEQ XEQM07	Command	XEQM07	XEQM07	Execute XEQM07 (GUDERM)
XEQ XEQM08	Command	XEQM08	XEQM08	Execute XEQM08 (PYTHAG)
XEQ XEQM09	Command	XEQM09	XEQM09	Execute XEQM09 (PLTPRIM)
XEQ XEQM10	Command	XEQM10	XEQM10	Execute XEQM10 (58TESTS)
XEQ XEQM11	Command	XEQM11	XEQM11	Execute XEQM11 (SINC_PI)
XEQ XEQM12	Command	XEQM12	XEQM12	Execute XEQM12 (BINET)
XEQ XEQM13	Command	XEQM13	XEQM13	Execute XEQM13 (TRAPZ)
XEQ XEQM14	Command	XEQM14	XEQM14	Execute XEQM14 (PLTFORM)
XEQ XEQM15	Command	XEQM15	XEQM15	Execute XEQM15 (X15)
XEQ XEQM16	Command	XEQM16	XEQM16	Execute XEQM16 (BINETF)
XEQ XEQM17	Command	XEQM17	XEQM17	Execute XEQM17 (RANDOM)
XEQ XEQM18	Command	XEQM18	XEQM18	Execute XEQM18 (TEST)
XEQM	MENU	XEQM	XEQM	Menu of predefined XEQC-functions
XI	Character	$\Xi$		Character $\Xi$ (926)
xi lowercase	Character	$\xi$		Character $\xi$ (958)
XOR	Function (dyadic)	XOR	XOR	Logical exclusive OR
Xor	Character	$\Upsilon$		Character $\Upsilon$ (8891)
xsum	Command	$x_{\text{sum}}$	$x_{\text{sum}}$	Return $\Sigma x$ and $\Sigma y$ in X and Y respectively ( $\Sigma_x ; \Sigma_y = (2 \text{ stack levels})$ )
xth root	Character	$\sqrt[x]{y}$		Character $\sqrt[x]{y}$ (8732)
xth root	Function (dyadic)	$\sqrt[x]{y}$	$\sqrt[x]{y}$	Xth root of Y
XXEQ	MENU	XXEQ	XXEQ	XXEQ menu (longer[XEQ] (KEY.LP ON))
Y	Character	Y		Character Y (89)
Y	User Program	Y		Y test program, preloaded from testPgms.bin
Y acute	Character	$\acute{Y}$		Character $\acute{Y}$ (221)
y acute lowercase	Character	$\acute{y}$		Character $\acute{y}$ (253)
Y circumflex	Character	$\hat{Y}$		Character $\hat{Y}$ (374)
y circumflex lowercase	Character	$\hat{y}$		Character $\hat{y}$ (375)
Y diaeresis	Character	$\ddot{Y}$		Character $\ddot{Y}$ (376)
y diaeresis lowercase	Character	$\ddot{y}$		Character $\ddot{y}$ (255)
y hat	Command	$\hat{y}$	$\hat{y}$	Estimation of y
y lowercase	Character	y		Character y (121)

FullName	Type	Label	Catalog	Description
y macron lowercase	Character	ȳ		Character ȳ (563)
y to the power x	Function (dyadic)	$y^x$	$y^x$	Raise value in the Y-register to the power in the X-register
y under root	Character	ȳ		Character ȳ (562)
yd. to m	Function (linked ; monadic)	yd.→m		Convert yard to meter (m:)
Year	Command	YEAR	YEAR	Year (of date)
Year month day	Setting (pgm)	YMD	YMD (*)	Date display mode YYYY-MM-DD (YYYY-MM-DD)
Year month day	Flag (system)	YMD		Date display mode YYYY-MM-DD (YYYY-MM-DD)
year to s	Function (linked ; monadic)	year→s		Convert year to second (s:)
Yen	Character	¥		Character ¥ (165)
Yes	Command	YES		Respond : Yes ([ENTER])
yīn to m	Function (linked ; monadic)	yīn→m		Convert yīn to meter (m:)
$Y_y(x)$	Command	$Y_y(x)$	$Y_y(x)$	Bessel function of the 2nd kind and order y
Z	Character	Z		Character Z (90)
Z	User Program	Z		Z test program, preloaded from testPgms.bin
Z acute	Character	Ẑ		Character Ẑ (377)
z acute lowercase	Character	ẑ		Character ẑ (378)
Z caron	Character	Ž		Character Ž (381)
z caron lowercase	Character	ž		Character ž (382)
Z dot	Character	Ẓ		Character Ẓ (379)
z dot lowercase	Character	ẓ		Character ẓ (380)
z lowercase	Character	z		Character z (122)
Z^Y modulo X	Function (triadic)	^MOD	^MOD	Z^Y modulo X
Zero	Constant (#78)	#		zero # = 0
ZETA	Character	Z		Character Z (918)
zeta lowercase	Character	ζ		Character ζ (950)
zhāng to m	Function (linked ; monadic)	zhāng→m		Convert zhāng to meter (m:)
Zoom (plot)	Command	ZOOM		Cycle through a preset list of zoom factors on x- and y-axis (-3 ; -2 ; -1 ; 0) (-3..-1 : zoom out ; 0 : normal)
Zoom y-axis	Command	ZOOMy		Cycle through a preset list of zoom factors on the y-axis (-1 ; 0 ; 1 ; 2 ; 3) (-1 : zoom out ; 0 : normal ; 1..3 : zoom in)
ZxY modulo X	Function (triadic)	xMOD	xMOD	Z x Y, modulo X
Γ function	Function (monadic)	Γ(x)	Γ(x)	Gamma function
Γ <sub>xy</sub>	Function (dyadic)	Γ <sub>xy</sub>	Γ <sub>xy</sub>	Upper incomplete Gamma function
γ <sub>xy</sub>	Function (dyadic)	γ <sub>xy</sub>	γ <sub>xy</sub>	Lower incomplete Gamma function
ζ(x)	Function (monadic)	ζ(x)	ζ(x)	Riemann's Zeta for real arguments
ΣV	User Program	ΣV		ΣV test program, preloaded from testPgms.bin
φ <sub>E</sub> (x)	Function (monadic)	φ <sub>E</sub> (x)	φ <sub>E</sub> (x)	Euler's totient function