

C47 Full names by category

Category	Type	FullName	Default	Label	Catalog	Extended description
Acceleration conversions	Function (monadic, linked)	ft/s ² to m/s ²		ft/s ² →m/s ²		Convert foot per second squared to metre per second squared (TI : m/s ² ;))
	Function (monadic, linked)	in/s ² to m/s ²		in/s ² →m/s ²		Convert inch per second squared to metre per second squared (TI : m/s ² ;))
	Function (monadic, linked)	m/s ² to ft/s ²		m/s ² →ft/s ²		Convert metre per second squared to foot per second squared (TI : ft/s ² ;))
	Function (monadic, linked)	m/s ² to in/s ²		m/s ² →in/s ²		Convert metre per second squared to inch per second squared (TI : in/s ² ;))
Angle conversions	Function (monadic, linked)	deg to grad		deg→grad		Convert degree to gradian (untagged) (TI : grad;))
	Function (monadic, linked)	deg to rad		deg→rad		Convert degree to radian (untagged) (TI : rad;))
	Function (monadic, linked)	grad to deg		grad→deg		Convert gradian to degree (untagged) (TI : deg;))
	Function (monadic, linked)	grad to rad		grad→rad		Convert gradian to radian (untagged) (TI : rad;))
	Function (monadic, linked)	rad to deg		rad→deg		Convert radian to degree (untagged) (TI : deg;))
	Function (monadic, linked)	rad to grad		rad→grad		Convert radian to gradian (untagged) (TI : grad;))
	Function (monadic, cyclic, no	Degrees-radians-gradians		DRG	DRG	Add ADM tag to untagged value in X, convert tagged value to degrees-radians-gradians (cyclic) (Info : Setting tag for complex X also sets POLAR mode for X according to ADM)
Angle tagging or conversion	Setting (pgm)	Set ADM to D.MS	OFF	D.MS	D.MS ○	Set Angular Display Mode to sexagesimal degrees mode (ADM = 1) (Info : SBI ¼ is shown if flag SBang and SBadm is SET)
	Setting (pgm)	Set ADM to DEG	ON	DEG	DEG ●	Set Angular Display Mode to degrees mode (ADM = 0) (Info : SBI ° is shown if flag SBadm is SET ; SBI ¼ is shown if flag SBang and SBadm is SET)
	Setting (pgm)	Set ADM to GRAD	OFF	GRAD	GRAD ○	Set Angular Display Mode to gradians mode (ADM = 4) (Info : SBI ¼ is shown if flag SBang and SBadm is SET)
	Setting (pgm)	Set ADM to MULπ	OFF	MULπ	MULπ ○	Set Angular Display Mode to multiple of pi radians mode (ADM = 3) (Info : SBI ¼ is shown if flag SBang and SBadm is SET)
	Setting (pgm)	Set ADM to RAD	OFF	RAD	RAD ○	Set Angular Display Mode to radians mode (ADM = 2) (Info : SBI ¼ is shown if flag SBang and SBadm is SET)
	Function (monadic)	Set D.MS tag or convert to D.MS		→D.MS	→D.MS	If untagged, set tag to D.MS (sexagesimal degrees) ; if tagged, convert X to D.MS ; does not change Angular Display Mode (Tag : °)")
	Function (monadic)	Set DEG tag or convert to DEG		→DEG	→DEG	If untagged, set tag to DEG (degrees) ; if tagged, convert X to DEG ; does not change Angular Display Mode (Tag : °)")
	Function (monadic)	Set GRAD tag or convert to GRAD		→GRAD	→GRAD	If untagged, set tag to GRAD (gradians) ; if tagged, convert X to GRAD ; does not change Angular Display Mode (Tag : °)")
	Function (monadic)	Set MULπ tag or convert to MULπ		→MULπ	→MULπ	If untagged, set tag to MULπ (multiple of pi radians) ; if tagged, convert X to MULπ ; does not change Angular Display Mode (Tag : °)")
	Function (monadic)	Set RAD tag or convert to RAD		→RAD	→RAD	If untagged, set tag to RAD (radians) ; if tagged, convert X to RAD ; does not change Angular Display Mode (Tag : °)")
	Command	Get angular display mode	0	ADM#	ADM#	Get angular display mode ; set interactively from menu MODE formats or from catalog (DEG ; D.MS ; RAD ; MULπ ; GRAD) and by ADM (Info : DEG : 0 ; D.MS : 1 ; RAD : 2 ; MULπ : 3 ; GRAD : 4)
	Command	Set angular display mode	0	ADM	ADM ₀	Set angular display mode to X ; also set interactively from menu MODE formats or from catalog (DEG ; D.MS ; RAD ; MULπ ; GRAD) (Info : DEG : 0 ; D.MS : 1 ; RAD : 2 ; MULπ : 3 ; GRAD : 4)
Angular/Speed conversions	MENU	Angle and angular velocity conversions		[Angle:]	Angle:	Convert between units of angle or angular velocity (Info : Flag CONV _{HP} may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> → ← <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Angle_page_1.pdf
	Function (monadic, linked)	deg/s to rad/s		deg/s→rad/s		Convert degree per second to radian per second (TI : rad/s;))
	Function (monadic, linked)	deg/s to RPM		deg/s→RPM		Convert degree per second to rotation per minute (TI : RPM;))
	Function (monadic, linked)	rad/s to deg/s		rad/s→deg/s		Convert radian per second to degree per second (TI : deg/s;))
	Function (monadic, linked)	rad/s to RPM		rad/s→RPM		Convert radian per second to rotation per minute (TI : RPM;))
	Function (monadic, linked)	RPM to deg/s		RPM→deg/s		Convert rotation per minute to degree per second (TI : deg/s;))
	Function (monadic, linked)	RPM to rad/s		RPM→rad/s		Convert rotation per minute to radian per second (TI : rad/s;))
Area conversions	MENU	Area conversions		[Area:]	Area:	Convert between units of area (Info : Flag CONV _{HP} may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> → ← <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Area_page_1.pdf
	Function (monadic, linked)	acre to ha		acre→ha		Convert acre to hectare (TI : ha;))
	Function (monadic, linked)	acre _{US} to ha		acre _{US} →ha		Convert US acre to hectare (TI : ha;))
	Function (monadic, linked)	ft ² to ha		ft ² →ha		Convert square foot to hectare (TI : ha;))
	Function (monadic, linked)	ft ² to m ²		ft ² →m ²		Convert square foot to square metre (TI : m ² ;))
	Function (monadic, linked)	ha to acre		ha→acre		Convert hectare to acre (TI : acre;))
	Function (monadic, linked)	ha to acre _{US}		ha→acre _{US}		Convert hectare to US acre (TI : acre _{US} ;))
	Function (monadic, linked)	ha to ft ²		ha→ft ²		Convert hectare to square foot (TI : ft ² ;))
	Function (monadic, linked)	ha to km ²		ha→km ²		Convert hectare to square kilometre (TI : km ² ;))
	Function (monadic, linked)	ha to m ²		ha→m ²		Convert hectare to square metre (TI : m ² ;))
	Function (monadic, linked)	in ² to mm ²		in ² →mm ²		Convert square inch to square millimetre (TI : mm ² ;))
	Function (monadic, linked)	km ² to ha		km ² →ha		Convert square kilometre to hectare (TI : ha;))

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, linked)	km ² to mi ²		km ² →mi ²		Convert square kilometre to square mile (TI : mi ²);
	Function (monadic, linked)	km ² to nmi ²		km ² →nmi ²		Convert square kilometre to square nautical mile (TI : nmi ²);
	Function (monadic, linked)	m ² to ft ²		m ² →ft ²		Convert square metre to square foot (TI : ft ²);
	Function (monadic, linked)	m ² to ha		m ² →ha		Convert square metre to hectare (TI : ha);
	Function (monadic, linked)	m ² to mü		m ² →mü		Convert square metre to mü (TI : mü);
	Function (monadic, linked)	mi ² to km ²		mi ² →km ²		Convert square mile to square kilometre (TI : km ²);
	Function (monadic, linked)	mm ² to in ²		mm ² →in ²		Convert square millimetre to square inch (TI : in ²);
	Function (monadic, linked)	mü to m ²		mü→m ²		Convert mü to square metre (TI : m ²);
	Function (monadic, linked)	nmi ² to km ²		nmi ² →km ²		Convert square nautical mile to square kilometre (TI : km ²);
Bits and bytes	Setting (pgm)	1's complement	OFF	1COMPL	1COMPL ○	Set 1's complement mode for Long integer (ISM = 1) (Info : <w> in SBI indicates word size (menu BITSET) ; SBI <w>:1 is shown if flag SBint is SET)
	Setting (pgm)	2's complement	ON	2COMPL	2COMPL ●	Set 2's complement mode for Long integer (ISM = 2) (Info : <w> in SBI indicates word size (menu BITSET) ; SBI <w>:2 is shown if flag SBint is SET)
	Setting (pgm)	BCD 10's complement	OFF	10CMPL	10CMPL ○	Set BCD 10's complement display of Long integer in #BASE Mode (Info : If menu BASE is open (#BASE Mode), BCD is displayed with subscript digits and followed by subscript _{bcd})
	Setting (pgm)	BCD 9's complement	OFF	9CMPL	9CMPL ○	Set BCD 9's complement display of Long integer in #BASE Mode (Info : If menu BASE is open (#BASE Mode), BCD is displayed with subscript digits and followed by subscript _{bcd})
	Setting (pgm)	BCD unsigned	ON	BCDUNS	BCDUNS ●	Set BCD unsigned display of Long integer in #BASE Mode (Info : If menu BASE is open (#BASE Mode), BCD is displayed with subscript digits and followed by subscript _{bcd})
	Setting (pgm)	Set word size	64	WSIZE	WSIZE ₆₄	Set word size in bits, for Short integer (8.02-8.16) (TAM : WSIZE __ TamNonReg menu menu ; Info : <m> in SBI indicates sign mode (menu BITSET) ; SBI 64:<m> is shown if flag SBint is SET)
	Setting (pgm)	Sign and mantissa	OFF	SIGNMT	SIGNMT ○	Set sign and mantissa mode for Short integer (ISM = -1) (Info : <w> in SBI indicates word size (menu BITSET) ; SBI <w>:s is shown if flag SBint is SET)
	Setting (pgm)	Unsigned	OFF	UNSIGN	UNSIGN ○	Set unsigned mode for Short integer (ISM = 0) (Info : <w> in SBI indicates word size (menu BITSET) ; SBI <w>:u is shown if flag SBint is SET)
	Setting	Binary coded decimal	OFF	BCD	BCD □	Switch on binary coded decimal (BCD) display of Long integer in #BASE Mode ; not programmable - in PEM, use flag BCD (Info : If menu BASE is open (#BASE Mode), BCD is displayed with subscript digits and followed by subscript _{bcd})
	Setting	Carry (bit)	OFF	CARRY	CARRY □	Switch on carry (bit) ; not programmable - in PEM, use flag CARRY (Info : SBI : is shown if flag SBoc is SET)
	Setting	Overflow	OFF	OVERFL	OVERFL □	Switch on overflow ; not programmable - in PEM, use flag OVERFL (Info : SBI : is shown if flag SBoc is SET)
	MENU (item)	Bit shift functions		[BITSHFT]	BITS↕2	Activate menu BITS page 2 to access bit shift functions (Info : Previews/OPENM as BITS↕2) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_BITSHFT_page_1.pdf
	MENU	Bit set functions		[BITSET]	BITSET	Bit set functions Docs: https://47calc.com/doc/C47/Menus/C47_Menu_BITSET_page_1.pdf
	MENU	Bitwise functions		[BITS]	BITS	Bitwise functions ; Keyboard g[4] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_BITS_page_1.pdf
	Function (monadic)	Arithmetic Shift Right		ASRn	ASRn	Arithmetic Shift Right with number of bits, preserving the sign of signed integers (trailing input) (TAM : ASRn __ TamNonReg menu)
	Function (monadic)	Flip bit		FB	FB	Flip bit n (TAM : FB __ TamNonReg menu)
	Function (monadic)	Left bit mask		MASKL	MASKL	Set left n bits to use as mask (TAM : MASKL __ TamNonReg menu)
	Function (monadic)	Left justify		LJ	LJ	Left justify (within word size) ; returns shift in X and result in Y
	Function (monadic)	Mirror bits		MIRROR	MIRROR	Flip bits
	Function (monadic)	NOT		NOT	NOT	Bitwise NOT
	Function (monadic)	Right bit mask		MASKR	MASKR	Set right n bits to use as mask (TAM : MASKR __ TamNonReg menu)
	Function (monadic)	Right justify		RJ	RJ	Right justify (within word size) ; returns shift in X and result in Y
	Function (monadic)	Rotate Left		RLn	RLn	Rotate Left with number of bits (trailing input) (TAM : RLn __ TamNonReg menu)
	Function (monadic)	Rotate Left (1)		RL	RL	Shortcut to Rotate Left (1 bit)
	Function (monadic)	Rotate Left through Carry		RLCn	RLCn	Rotate Left through Carry with number of bits (trailing input) (TAM : RLCn __ TamNonReg menu)
	Function (monadic)	Rotate Left through Carry (1)		RLC	RLC	Shortcut to Rotate Left through Carry (1 bit)
	Function (monadic)	Rotate right		RRn	RRn	Rotate right with number of bits (trailing input) (TAM : RRn __ TamNonReg menu)
	Function (monadic)	Rotate right (1)		RR	RR	Shortcut to Rotate right (1 bit)
	Function (monadic)	Rotate Right through Carry		RRCn	RRCn	Rotate Right through Carry with number of bits (trailing input) (TAM : RRCn __ TamNonReg menu)
	Function (monadic)	Rotate Right through Carry (1)		RRC	RRC	Shortcut to Rotate Right through Carry (1 bit)
	Function (monadic)	Unzip		UNZIP	UNZIP	De-interleave (unzip) bits of X into Y and X, using most significant bit for Y
	Function (dyadic)	AND		AND	AND	Bitwise AND
	Function (dyadic)	Exclusive NOR		XNOR	XNOR	Logical exclusive NOR (bitwise)
	Function (dyadic)	NOR		NOR	NOR	Logical not OR (bitwise)
	Function (dyadic)	Not AND		NAND	NAND	Logical not AND (bitwise)
	Function (dyadic)	OR		OR	OR	Bitwise OR
	Function (dyadic)	XOR		XOR	XOR	Bitwise XOR (exclusive OR)
	Function (dyadic)	Zip		ZIP	ZIP	Interleave (zip) bits of Y and X, starting with most significant bit of Y

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag	Binary coded decimal	CLEAR	BCD		SET for binary coded decimal (BCD) display of Long integer in #BASE Mode; CLEAR for binary coded decimal display inactive (Info : If menu BASE is open (#BASE Mode), BCD is displayed with subscript digits and followed by subscript <i>bcd</i>)
	Flag	Carry (bit)	CLEAR	CARRY		SET for carry (bit); CLEAR for carry (bit) inactive (Info : SBI _e is shown if flag SBoc is SET)
	Flag	Overflow	CLEAR	OVERFL		SET for overflow; CLEAR for no overflow (Info : SBI _o is shown if flag SBoc is SET)
	Command (logical)	Bit clear?		BC?	BC?	Test bit n is clear (TAM : BC? ₋₋₋ TamNonReg menu)
	Command	Get word size	64	WSIZE#	WSIZE#	Get word size in bits, for Short integer (8.02-8.16) (TI : Bits = ; Info : set by WSIZE)
Catalog	MENU (ASM)	All variables		[ALL]	ALLVARS	Auto-generated catalog of all variables (Info : Matrix variables STATS and HISTO are also included in this menu, if created) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ALLVARS_page_1.pdf
	MENU (ASM)	Constants		[CNST]	CNST	Important scientific and technical constants (Info : In Program Entry Mode, constants selected from the menu are stored in programs as "# ", the reference number, the label and the abbreviated name; in the equation editor the label is simply preceded by "#"; Type characters 1-2 to search; TI : <constant abbreviation> = <value> ; Ref : https://47calc.com/doc/C47/Reference/C47_Constants.pdf ; Keyboard g[+] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CNST_page_1.pdf
	MENU (ASM)	Functions		[FCNS]	FCNS	Catalog of all calculator functions (Info : Type characters 1-2 to search; Ref : https://47calc.com/doc/C47/Reference/C47_CAT-FCNS.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FCNS_page_1.pdf
	MENU (ASM)	Functions (equation)		[FCNS]	FCNS _e	Catalog of all calculator functions available in Equation Input Mode (Info : Type characters 1-2 to search) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FCNS_EQN_page_1.pdf
	MENU (ASM)	Menus		[MENUS]	MENUS	Catalog of all menus (including user defined menus) (Info : Type characters 1-2 to search; Ref : https://47calc.com/doc/C47/Reference/C47_CAT-MENUS.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MENUS_page_1.pdf
	MENU (ASM)	Programs		[PROGS]	PROGS	Auto-generated catalog of programs Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PROGS_page_1.pdf
	MENU	Angle variables		[ANGLES]	ANGLES	Auto-generated catalog of variables of the specified datatype : Angle (1.1-1.5) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ANGLES_page_1.pdf
	MENU	Catalog		[CAT]	CATALOG	Catalog of all items (functions, characters, programs, variables, menus) (Info : Press [EXIT] to exit catalogs) ; Keyboard f[+] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CAT_page_1.pdf
	MENU	Catalog (alpha)		[CAT]	CAT _α	Catalog of items available in Alpha Input Mode (Info : CAT [ALPHA] is available on keyboard alpha f[+]) ; Keyboard alpha f[+] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CAT_ALPHA_page_1.pdf
	MENU	Catalog (equation)		[CAT]	CAT _e	Catalog of items available in Equation Input Mode (Info : CAT [EQN] is available on keyboard f[+] in EIM) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CAT_EQN_page_1.pdf
	MENU	Characters		[CHARS]	CHARS	Access to all character submenus (Mya ; A...Ω/α...ω ; αMath ; αMisc ; αINTL/aintl) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CHARS_page_1.pdf
	MENU	Complex variables		[CPXS]	CPXS	Auto-generated catalog of variables of the specified datatype : Complex number (2.0-2.5) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CPXS_page_1.pdf
	MENU	Config variables		[CONFIGS]	CONFIGS	Auto-generated catalog of variables of the specified datatype : Config (9) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CONFIGS_page_1.pdf
	MENU	Date variables		[DATES]	DATES	Auto-generated catalog of variables of the specified datatype : Date (4) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DATES_page_1.pdf
	MENU	Long integer variables		[L.INTS]	L.INTS	Auto-generated catalog of variables of the specified datatype : Long integer (0) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_L-INTS_page_1.pdf
	MENU	Number variables		[NUMBRS]	NUMBRS	Auto-generated catalog of variables of the specified datatype : Number (0 ; 1.0-1.5 ; 2.0-2.5) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_NUMBRS_page_1.pdf
	MENU	Real matrix or vector variables		[MATRS]	MATRS	Auto-generated catalog of variables of the specified datatype : Real matrix or vector (6) or Complex matrix or vector (7.n) (Info : Matrix variables STATS and HISTO are also included in this menu, if created) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MATRS_page_1.pdf
	MENU	Real variables		[REALS]	REALS	Auto-generated catalog of variables of the specified datatype : Real number (1.0-1.5) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_REALS_page_1.pdf
	MENU	Short integer variables		[S.INTS]	S.INTS	Auto-generated catalog of variables of the specified datatype : Short integer (8.02-8.16) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_S-INTS_page_1.pdf
	MENU	Text string variables		[STRINGS]	STRINGS	Auto-generated catalog of variables of the specified datatype : Text string (5) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_STRINGS_page_1.pdf
	MENU	Time variables		[TIMES]	TIMES	Auto-generated catalog of variables of the specified datatype : Time (3) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TIMES_page_1.pdf

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU	Variables		[VARS]	VARS	Auto-generated variable catalogs Docs: https://47calc.com/doc/C47/Menus/C47_Menu_VARS_page_1.pdf
	Browser	Keyboard map		KEYMAP	KEYMAP	Browse keyboard map (default USER mode) ; also showing special NORMAL hotkey assignment from menu Σ + KEY ; if KEYMAP is activated when in Alpha Input Mode or Equation Input Mode, the alpha keyboard mappings are shown only (default NORMAL mode, starting with alpha g-shift) (Info : Press and hold key [.] to view USER or NORMAL (STD) keys ; browse using [▲] and [▼] ; press [EXIT] to exit Keyboard map ; Ref : https://47calc.com/doc/C47/Reference/C47_KEYMAP.pdf)
Characters	MENU	Alpha international		[aINTL]	aINTL	Uppercase international characters (Info : Lowercase menu via [▼] ; single character selection) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_alphaiNTL_page_1.pdf
	MENU	alpha international		[aintl]	aintl	Lowercase international characters (Info : Uppercase menu via [▲] ; single character selection) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_alphaintl-lower_page_1.pdf
	MENU	Alpha Math		[aMath]	aMath	Mathematical symbols (Info : Single character selection) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_alphaMATH_page_1.pdf
	MENU	Alpha miscellaneous		[aMisc]	aMisc	Special characters (Info : Single character selection) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_alphaMisc_page_1.pdf
	MENU	ALPHA...OMEGA		[A...Ω]	A...Ω	Uppercase Greek characters (Info : Lowercase menu via [▼] ; single character selection) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Alpha-Omega_page_1.pdf
	MENU	alpha...omega		[a...ω]	a...ω	Lowercase Greek characters (Info : Uppercase menu via [▲] ; single character selection) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_alpha-omega-lower_page_1.pdf
	Character	Sun		☉		Character ☉ (Fontcode : 2299 ₁₆)
	Character	a		Ⓐ		Character Ⓐ (Fontcode : 0061 ₁₆) ; Keyboard alpha f[Σ+]
	Character	A accute		Ā		Character Ā (Fontcode : 00C1 ₁₆ ; Info : press [▼] for ā at [aintl.1]F2)
	Character	a accute		ā		Character ā (Fontcode : 00E1 ₁₆ ; Info : press [▲] for Ā at [aINTL.1]F2)
	Character	A breve		Ă		Character Ă (Fontcode : 0102 ₁₆ ; Info : press [▼] for ă at [aintl.1]F3)
	Character	a breve		ă		Character ă (Fontcode : 0103 ₁₆ ; Info : press [▲] for Ă at [aINTL.1]F3)
	Character	A circumflex		Â		Character Â (Fontcode : 00C2 ₁₆ ; Info : press [▼] for â at [aintl.1]F3)
	Character	a circumflex		â		Character â (Fontcode : 00E2 ₁₆ ; Info : press [▲] for Â at [aINTL.1]F3)
	Character	A diaeresis		Ä		Character Ä (Fontcode : 00C4 ₁₆ ; Info : press [▼] for ä at [aintl.1]F5)
	Character	a diaeresis		ä		Character ä (Fontcode : 00E4 ₁₆ ; Info : press [▲] for Ä at [aINTL.1]F5)
	Character	A grave		À		Character À (Fontcode : 00C0 ₁₆ ; Info : press [▼] for à at [aintl.1]F1)
	Character	a grave		à		Character à (Fontcode : 00E0 ₁₆ ; Info : press [▲] for À at [aINTL.1]F1)
	Character	A macron		Ā		Character Ā (Fontcode : 01F1 ₁₆ ; Info : press [▼] for ā at [aintl.1]F2)
	Character	a macron		ā		Character ā (Fontcode : 0101 ₁₆ ; Info : press [▲] for Ā at [aINTL.1]F2)
	Character	A ogonek		Ą		Character ą (Fontcode : 0104 ₁₆ ; Info : press [▼] for ą at [aintl.1]F4)
	Character	a ogonek		ą		Character ą (Fontcode : 0105 ₁₆ ; Info : press [▲] for Ą at [aINTL.1]F4)
	Character	A ring		Å		Character Å (Fontcode : 00C5 ₁₆ ; Info : press [▼] for å at [aintl.1]F6)
	Character	a ring		å		Character å (Fontcode : 00E5 ₁₆ ; Info : press [▲] for Å at [aINTL.1]F6)
	Character	A tilde		Ȧ		Character Ȧ (Fontcode : 00C3 ₁₆ ; Info : press [▼] for ȧ at [aintl.1]F4)
	Character	a tilde		ȧ		Character ȧ (Fontcode : 00E3 ₁₆ ; Info : press [▲] for Ȧ at [aINTL.1]F4)
	Character	AC current		Ⓐ		Character Ⓐ (Fontcode : 23E6 ₁₆)
	Character	AE		Æ		Character Æ (Fontcode : 00C6 ₁₆ ; Info : press [▼] for æ at [aintl.1]F1)
	Character	ae		æ		Character æ (Fontcode : 00E6 ₁₆ ; Info : press [▲] for Æ at [aINTL.1]F1)
	Character	Almost equal		≈		Character ≈ (Fontcode : 2248 ₁₆)
	Character	ALPHA		Α		Character Α (Fontcode : 0391 ₁₆ ; Info : press [▼] for α at [a...ω.1]F1)
	Character	alpha		α		Character α (Shortcut : alpha g[XEQ] ; Info : Fontcode : 03B1 ₁₆ ; press [▲] for Α at [A...Ω.1]F1)
	Character	alpha tonos		ά		Character ά (Fontcode : 03AC ₁₆)
	Character	Ampersand		&		Character & (Fontcode : 0026 ₁₆)
	Character	And		Λ		Character Λ (Fontcode : 2227 ₁₆)
	Character	Angle		∠		Character ∠ (Fontcode : 2220 ₁₆)
	Character	Apostrophe		'		Character ' (Fontcode : 0027 ₁₆)
	Character	Asterisk		*		Character * (Fontcode : 002A ₁₆)
	Character	Asymptotically equal to		≈		Character ≈ (Fontcode : 2243 ₁₆)
	Character	At		@		Character @ (Fontcode : 0040 ₁₆)
	Character	b		Ⓑ		Character Ⓑ (Fontcode : 0062 ₁₆) ; Keyboard alpha f[1/x]
	Character	Back slash		\		Character \ (Fontcode : 005C ₁₆)
	Character	Battery		🔋		Character 🔋 (Fontcode : 242A ₁₆)
	Character	Battery half		🔋		Character 🔋 (Fontcode : 2435 ₁₆)
	Character	BETA		Β		Character Β (Fontcode : 0392 ₁₆ ; Info : press [▼] for β at [a...ω.1]F2)
	Character	beta		β		Character β (Fontcode : 03B2 ₁₆ ; Info : press [▲] for Β at [A...Ω.1]F2)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	Binary 0		0		Character 0 (Fontcode : 220E ₁₆)
	Character	Binary 1		1		Character 1 (Fontcode : 2027 ₁₆)
	Character	BST		Δ		Character Δ (Fontcode : 21C9 ₁₆)
	Character	Bullet		•		Character • (Fontcode : 2219 ₁₆)
	Character	c		c		Character c (Fontcode : 0063 ₁₆) ; Keyboard alpha f[√x]
	Character	c accute		Ċ		Character Ċ (Fontcode : 0106 ₁₆ ; Info : press [▼] for ċ at [aintl.1]fF6)
	Character	c accute		ċ		Character ċ (Fontcode : 0107 ₁₆ ; Info : press [▲] for Ċ at [aintl.1]fF6)
	Character	c caron		Ĉ		Character Ĉ (Fontcode : 010C ₁₆ ; Info : press [▼] for ĉ at [aintl.1]gF1)
	Character	c caron		ĉ		Character ĉ (Fontcode : 010D ₁₆ ; Info : press [▲] for Ĉ at [aintl.1]gF1)
	Character	C cedilla		Ç		Character Ç (Fontcode : 00C7 ₁₆ ; Info : press [▼] for ç at [aintl.1]fF5)
	Character	c cedilla		ç		Character ç (Fontcode : 00E7 ₁₆ ; Info : press [▲] for Ç at [aintl.1]fF5)
	Character	Carriage return		↵		Character ↵ (Shortcut : alpha g[ENTER] ; alpha longer [ENTER] ; EIM longer [ENTER] ; Info : Fontcode : 21B5 ₁₆)
	Character	CB_OFF		□		Character □ (Fontcode : 2610 ₁₆)
	Character	CB_ON		▣		Character ▣ (Fontcode : 2612 ₁₆)
	Character	Cent		¢		Character ¢ (Fontcode : 00A2 ₁₆)
	Character	Check mark		✓		Character ✓ (Fontcode : 2713 ₁₆)
	Character	CHI		X		Character X (Fontcode : 03A7 ₁₆ ; Info : press [▼] for x at [a...w.2]fF1)
	Character	chi		x		Character x (Fontcode : 03C7 ₁₆ ; Info : press [▲] for X at [A...Ω.2]fF1)
	Character	Colon		:		Character : ; in the equation editor, the colon can be used to name an expression (<name> : <expression>) (Shortcut : alpha [0] ; Fontcode : 003A ₁₆)
	Character	Colon equals		:=		Character := (Fontcode : 2254 ₁₆)
	Character	Comma		,		Character , (Fontcode : 002C ₁₆) ; Keyboard alpha [.]
	Character	Complec C		Ĉ		Character Ĉ (Fontcode : 2102 ₁₆)
	Character	Complement		Ċ		Character Ċ (Fontcode : 2201 ₁₆)
	Character	Complex i		<i>i</i>		Character <i>i</i> (Info : Fontcode : 2148 ₁₆)
	Character	Complex j		<i>j</i>		Character <i>j</i> (Info : Fontcode : 2149 ₁₆)
	Character	Contains as member		⊃		Character ⊃ (Fontcode : 220B ₁₆)
	Character	Contour integral		∮		Character ∮ (Fontcode : 222E ₁₆)
	Character	Convert to xxx		⇒		Character ⇒ (Fontcode : 21D2 ₁₆)
	Character	Corresponds to		≡		Character ≡ (Fontcode : 2258 ₁₆)
	Character	Cross		×		Character × (Shortcut : alpha g[x] ; Info : Fontcode : 00D7 ₁₆)
	Character	Cursor		_		Character _ (Fontcode : 2427 ₁₆)
	Character	Cyclic		↻		Character ↻ (Fontcode : 267A ₁₆)
	Character	d		d		Character d (Fontcode : 0064 ₁₆) ; Keyboard alpha f[LOG]
	Character	d apostrophe		d'		Character d' (Fontcode : 010F ₁₆)
	Character	D caron		Ď		Character Ď (Fontcode : 010E ₁₆)
	Character	D stroke		Đ		Character Đ (Fontcode : 0110 ₁₆ ; Info : press [▼] for đ at [aintl.1]gF3)
	Character	d stroke		đ		Character đ (Fontcode : 0111 ₁₆ ; Info : press [▲] for Đ at [aintl.1]gF3)
	Character	Date		D		Character D (Fontcode : 2145 ₁₆)
	Character	DC current		—		Character — (Fontcode : 2393 ₁₆)
	Character	Degree sign		°		Character ° (Fontcode : 00B0 ₁₆)
	Character	DELTA		Δ		Character Δ (Shortcut : alpha g[RCL] ; Info : Fontcode : 0394 ₁₆ ; press [▼] for δ at [a...w.1]F4)
	Character	delta		δ		Character δ (Fontcode : 03B4 ₁₆ ; Info : press [▲] for Δ at [A...Ω.1]F4)
	Character	DIA_OFF		◊		Character ◊ (Fontcode : 25C7 ₁₆)
	Character	DIA_ON		◈		Character ◈ (Fontcode : 25C8 ₁₆)
	Character	DIGAMMA		Ɔ		Character Ɔ (Fontcode : 03DC ₁₆ ; Info : press [▼] for Ɔ at [a...w.1]F6)
	Character	digamma		Ɔ		Character Ɔ (Fontcode : 03DD ₁₆ ; Info : press [▲] for Ɔ at [A...Ω.1]F6)
	Character	Divides		∣		Character ∣ (Fontcode : 2223 ₁₆)
	Character	Does not contain as member		⊄		Character ⊄ (Fontcode : 220C ₁₆)
	Character	Does not divide		∤		Character ∤ (Fontcode : 2224 ₁₆)
	Character	Dollar		\$		Character \$ (Fontcode : 0024 ₁₆)
	Character	Double high-reversed-9 quotation mark		"		Character " (Fontcode : 201F ₁₆)
	Character	Double integral		∫		Character ∫ (Fontcode : 222C ₁₆)
	Character	Double low-9 quotation mark		„		Character „ (Fontcode : 201E ₁₆)
	Character	Down arrow		↓		Character ↓ (Fontcode : 21E3 ₁₆)
	Character	Down block arrow		⇓		Character ⇓ (Fontcode : 2B63 ₁₆)
	Character	Down hollow arrow		⇩		Character ⇩ (Fontcode : 21E9 ₁₆)
	Character	DP decimal separator		,		Character , (Fontcode : 2429 ₁₆)
	Character	DP decimal separator		,		Character , (Fontcode : 2428 ₁₆)
	Character	e		e		Character e (Fontcode : 0065 ₁₆) ; Keyboard alpha f[LN]
	Character	E accute		É		Character É (Fontcode : 00C9 ₁₆ ; Info : press [▼] for é at [aintl.1]gF5)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	e acute		é		Character é (Fontcode : 00E9 ₁₆ ; Info : press [▲] for É at [αINTL.1] gF5)
	Character	E breve		Ê		Character Ê (Fontcode : 0114 ₁₆ ; Info : press [▼] for ê at [αintl.2] F3)
	Character	e breve		ê		Character ê (Fontcode : 0115 ₁₆ ; Info : press [▲] for Ê at [αINTL.2] F3)
	Character	E caron		Ě		Character Ě (Fontcode : 011A ₁₆ ; Info : press [▼] for ě at [αintl.2] F6)
	Character	e caron		ě		Character ě (Fontcode : 011B ₁₆ ; Info : press [▲] for Ě at [αINTL.2] F6)
	Character	E circumflex		Ê		Character Ê (Fontcode : 00CA ₁₆ ; Info : press [▼] for ê at [αintl.1] gF8)
	Character	e circumflex		ê		Character ê (Fontcode : 00EA ₁₆ ; Info : press [▲] for Ê at [αINTL.1] gF6)
	Character	E dieresis		Ë		Character Ë (Fontcode : 00CB ₁₆ ; Info : press [▼] for ë at [αintl.2] F1)
	Character	e dieresis		ë		Character ë (Fontcode : 00EB ₁₆ ; Info : press [▲] for Ë at [αINTL.2] F1)
	Character	E dot		Ė		Character Ė (Fontcode : 0116 ₁₆ ; Info : press [▼] for ę at [αintl.2] F4)
	Character	e dot		ę		Character ę (Fontcode : 0117 ₁₆ ; Info : press [▲] for Ė at [αINTL.2] F4)
	Character	E grave		Ė		Character Ė (Fontcode : 00C8 ₁₆ ; Info : press [▼] for è at [αintl.1] gF4)
	Character	e grave		è		Character è (Fontcode : 00E8 ₁₆ ; Info : press [▲] for Ê at [αINTL.1] gF4)
	Character	E macron		Ē		Character Ē (Fontcode : 0112 ₁₆ ; Info : press [▼] for ē at [αintl.2] F2)
	Character	e macron		ē		Character ē (Fontcode : 0113 ₁₆ ; Info : press [▲] for Ē at [αINTL.2] F2)
	Character	E ogonek		Ę		Character Ę (Fontcode : 0118 ₁₆ ; Info : press [▼] for ę at [αintl.2] F5)
	Character	e ogonek		ę		Character ę (Fontcode : 0119 ₁₆ ; Info : press [▲] for Ę at [αINTL.2] F5)
	Character	eEuler		ℰ		Character ℰ (Shortcut : alpha f[-] ; Info : Fontcode : 2147 ₁₆)
	Character	Element of		∈		Character ∈ (Fontcode : 2208 ₁₆)
	Character	Ellipsis		...		Character ... (Fontcode : 2026 ₁₆)
	Character	Empty set		∅		Character ∅ (Fontcode : 2205 ₁₆)
	Character	EPSILON		Ε		Character Ε (Fontcode : 0395 ₁₆ ; Info : press [▼] for ε at [α...ω.1] F5)
	Character	epsilon		ε		Character ε (Fontcode : 03B5 ₁₆ ; Info : press [▲] for Ε at [Α...Ω.1] F5)
	Character	epsilon tolos		ϵ		Character ϵ (Fontcode : 03AD ₁₆)
	Character	Estimates		≐		Character ≐ (Fontcode : 2259 ₁₆)
	Character	ETA		Η		Character Η (Fontcode : 0397 ₁₆ ; Info : press [▼] for η at [α...ω.1] fF2)
	Character	eta		η		Character η (Fontcode : 03B7 ₁₆ ; Info : press [▲] for Η at [Α...Ω.1] fF2)
	Character	eta tolos		ῆ		Character ῆ (Fontcode : 03AE ₁₆)
	Character	ETH		Ð		Character Ð (Fontcode : 00D0 ₁₆)
	Character	eth		ð		Character ð (Fontcode : 00F0 ₁₆)
	Character	Euro		€		Character € (Fontcode : 20AC ₁₆)
	Character	exponent mark		ⁱ		Character ⁱ (Fontcode : 247D ₁₆)
	Character	Exponent sign (AIM)		<E>		Character Ε (displays as outline E in numeric font) (Shortcut : alpha g[E] ; Info : Fontcode : 2073 ₁₆)
	Character	f		f		Character f (Fontcode : 0066 ₁₆) ; Keyboard alpha f [XEQ]
	Character	final sigma		ς		Character ς (Fontcode : 03C2 ₁₆ ; Info : press [▲] for Σ at [Α...Ω.2] F3)
	Character	For all		∀		Character ∀ (Fontcode : 2C6F ₁₆ ; exported as 2200 ₁₆)
	Character	Fraction 1/2		½		Character ½ (Fontcode : 00BD ₁₆)
	Character	Fraction 1/4		¼		Character ¼ (Fontcode : 00BC ₁₆)
	Character	Full stop		.		Character . (Shortcut : alpha g[.] ; Info : Fontcode : 002E ₁₆)
	Character	G		G		Character G (Fontcode : 0047 ₁₆) ; Keyboard alpha [ST0]
	Character	g		g		Character g (Fontcode : 0067 ₁₆) ; Keyboard alpha f [ST0]
	Character	G breve		Ġ		Character Ġ (Fontcode : 011E ₁₆ ; Info : press [▼] for ġ at [αintl.2] fF1)
	Character	g breve		ġ		Character ġ (Fontcode : 011F ₁₆ ; Info : press [▲] for Ġ at [αINTL.2] fF1)
	Character	GAMMA		Γ		Character Γ (Fontcode : 0393 ₁₆ ; Info : press [▼] for γ at [α...ω.1] F3)
	Character	gamma		γ		Character γ (Fontcode : 03B3 ₁₆ ; Info : press [▲] for Γ at [Α...Ω.1] F3)
	Character	Greater or equal than		≥		Character ≥ (Fontcode : 2265 ₁₆)
	Character	Greater than		>		Character > (Fontcode : 003E ₁₆)
	Character	H		H		Character H (Fontcode : 0048 ₁₆) ; Keyboard alpha [RCL]
	Character	h		h		Character h (Fontcode : 0068 ₁₆) ; Keyboard alpha f [RCL]
	Character	h stroke		ħ		Character ħ (Fontcode : 0127 ₁₆)
	Character	Hourglass		⌚		Character ⌚ (Fontcode : 231B ₁₆)
	Character	HP42S Line Feed picture		␣		Character ␣ (Fontcode : 240A ₁₆)
	Character	HP42S Small black square		■		Character ■ (Fontcode : 25AA ₁₆)
	Character	I		I		Character I (Fontcode : 0049 ₁₆) ; Keyboard alpha [R+]
	Character	i		i		Character i (Fontcode : 0069 ₁₆) ; Keyboard alpha f [R+]
	Character	I acute		Í		Character Í (Fontcode : 00CD ₁₆ ; Info : press [▼] for í at [αintl.2] fF3)
	Character	i acute		í		Character í (Fontcode : 00ED ₁₆ ; Info : press [▲] for Í at [αINTL.2] fF3)
	Character	I breve		İ		Character İ (Fontcode : 012C ₁₆ ; Info : press [▼] for ĩ at [αintl.2] gF1)
	Character	i breve		ï		Character ï (Fontcode : 012D ₁₆ ; Info : press [▲] for İ at [αINTL.2] gF1)
	Character	I circumflex		Î		Character Î (Fontcode : 00CE ₁₆ ; Info : press [▼] for î at [αintl.2] fF4)
	Character	i circumflex		î		Character î (Fontcode : 00EE ₁₆ ; Info : press [▲] for Î at [αINTL.2] fF4)
	Character	I dieresis		Ï		Character Ï (Fontcode : 00CF ₁₆ ; Info : press [▼] for ï at [αintl.2] fF5)
	Character	i dieresis		ï		Character ï (Fontcode : 00EF ₁₆ ; Info : press [▲] for İ at [αINTL.2] fF5)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	I dot		İ		Character İ (Fontcode : 0130 ₁₆ ; Info : press [▼] for İ at [aintl.2] gF3 (not Lowercase of i))
	Character	i dotless		ı		Character ı (Fontcode : 0131 ₁₆ ; Info : press [▲] for İ at [aintl.2] gF3 (not UPPERcase of i))
	Character	i grave		ï		Character ï (Fontcode : 00CC ₁₆ ; Info : press [▼] for İ at [aintl.2] fF2)
	Character	i grave		ì		Character ì (Fontcode : 00EC ₁₆ ; Info : press [▲] for İ at [aintl.2] fF2)
	Character	i macron		ĩ		Character ĩ (Fontcode : 012A ₁₆ ; Info : press [▼] for İ at [aintl.2] fF6)
	Character	i macron		Ĳ		Character Ĳ (Fontcode : 012B ₁₆ ; Info : press [▲] for İ at [aintl.2] fF6)
	Character	i ogonek		į		Character į (Fontcode : 012E ₁₆ ; Info : press [▼] for İ at [aintl.2] gF2)
	Character	i ogonek		ı̇		Character ı̇ (Fontcode : 012F ₁₆ ; Info : press [▲] for İ at [aintl.2] gF2)
	Character	Identical to		≡		Character ≡ (Fontcode : 2261 ₁₆)
	Character	Increment		Δ		Character Δ (Fontcode : 2206 ₁₆)
	Character	Infinity		∞		Character ∞ (Fontcode : 221E ₁₆)
	Character	Integer Z		ℤ		Character ℤ (Fontcode : 2124 ₁₆)
	Character	Intersection		∩		Character ∩ (Fontcode : 2229 ₁₆)
	Character	Inverted exclamation mark		¡		Character ¡ (Fontcode : 00A1 ₁₆)
	Character	Inverted question mark		¿		Character ¿ (Fontcode : 00BF ₁₆)
	Character	IOTA		ι		Character ι (Fontcode : 0399 ₁₆ ; Info : press [▼] for ι at [α...ω.1] fF4)
	Character	iota		ı		Character ı (Fontcode : 03B9 ₁₆ ; Info : press [▲] for İ at [A...Ω.1] fF4)
	Character	IOTA dialytika		ϊ		Character ϐ (Fontcode : 03AA ₁₆ ; Info : press [▼] for ι at [α...ω.3] F1)
	Character	iota dialytika		ϐ		Character ϐ (Fontcode : 03CA ₁₆ ; Info : press [▲] for İ at [A...Ω.3] F1)
	Character	iota dialytika tonos		ϊ̇		Character ϐ̇ (Fontcode : 0390 ₁₆)
	Character	iota tonos		ı̇		Character ı̇ (Fontcode : 03AF ₁₆)
	Character	Irrational I		ℐ		Character ℐ (Fontcode : 2110 ₁₆)
	Character	J		Ј		Character Ј (Fontcode : 004A ₁₆) ; Keyboard alpha [SIN]
	Character	j		Ƶ		Character Ƶ (Fontcode : 006A ₁₆) ; Keyboard alpha f [SIN]
	Character	j circumflex		ĵ		Character ĵ (Fontcode : 0135 ₁₆)
	Character	K		Ʒ		Character K (Fontcode : 004B ₁₆) ; Keyboard alpha [COS]
	Character	k		ƶ		Character k (Fontcode : 006B ₁₆) ; Keyboard alpha f [COS]
	Character	k circumflex		ķ		Character ķ (Fontcode : 0137 ₁₆)
	Character	KAPPA		Ʒ		Character K (Fontcode : 039A ₁₆ ; Info : press [▼] for κ at [α...ω.1] fF5)
	Character	kappa		κ		Character κ (Fontcode : 03BA ₁₆ ; Info : press [▲] for K at [A...Ω.1] fF5)
	Character	KOPPA		Ʒ		Character Ʒ (Fontcode : 03D8 ₁₆ ; Info : press [▼] for ϱ at [α...ω.1] gF6)
	Character	koppa		ϱ		Character ϱ (Fontcode : 03D9 ₁₆ ; Info : press [▲] for Ʒ at [A...Ω.1] gF6)
	Character	L		ℒ		Character L (Fontcode : 004C ₁₆) ; Keyboard alpha [TAN]
	Character	l		ℓ		Character l (Fontcode : 006C ₁₆) ; Keyboard alpha f [TAN]
	Character	L accute		Ł		Character Ł (Fontcode : 0139 ₁₆ ; Info : press [▼] for ł at [aintl.2] gF4)
	Character	l accute		ł		Character ł (Fontcode : 013A ₁₆ ; Info : press [▲] for Ł at [aintl.2] gF4)
	Character	L apostrophe		Ł		Character Ł (Fontcode : 013D ₁₆ ; Info : press [▼] for ł at [aintl.2] gF5)
	Character	l apostrophe		ł		Character ł (Fontcode : 013E ₁₆ ; Info : press [▲] for Ł at [aintl.2] gF5)
	Character	l stoke		ℓ		Character ℓ (Fontcode : 0142 ₁₆ ; Info : press [▲] for Ł at [aintl.2] gF6)
	Character	L stroke		ℓ		Character ℓ (Fontcode : 0141 ₁₆ ; Info : press [▼] for ł at [aintl.2] gF6)
	Character	LAMDA		Λ		Character Λ (Fontcode : 039B ₁₆ ; Info : press [▼] for λ at [α...ω.1] fF6)
	Character	lamda		λ		Character λ (Fontcode : 03BB ₁₆ ; Info : press [▲] for Λ at [A...Ω.1] fF6)
	Character	Left arrow		←		Character ← (Fontcode : 21E0 ₁₆)
	Character	Left block arrow		⇐		Character ⇐ (Fontcode : 2860 ₁₆)
	Character	Left curly bracket		{		Character { (Fontcode : 007B ₁₆)
	Character	Left double quotation mark		“		Character “ (Fontcode : 201C ₁₆)
	Character	Left French quotation mark		«		Character « (Fontcode : 00AB ₁₆)
	Character	Left righr double arrow		⇐		Character ⇐ (Fontcode : 21D4 ₁₆)
	Character	Left right arrow		↔		Character ↔ (Shortcut : alpha g[x2y] ; Info : Fontcode : 21C4 ₁₆)
	Character	Left right arrow		⇔		Character ⇔ (Fontcode : 2194 ₁₆)
	Character	Left single quotation mark		’		Character ’ (Fontcode : 2018 ₁₆)
	Character	Left square bracket		[		Character [(Fontcode : 005B ₁₆)
	Character	less or equal than		≤		Character ≤ (Fontcode : 2264 ₁₆)
	Character	less than		<		Character < (Fontcode : 003C ₁₆)
	Character	List		≡		Character ≡ (Fontcode : 21CC ₁₆)
	Character	Litre		ℓ		Character ℓ (Fontcode : 2113 ₁₆)
	Character	M		Ʒ		Character M (Fontcode : 004D ₁₆) ; Keyboard alpha [x2y]
	Character	m		ƶ		Character m (Fontcode : 006D ₁₆) ; Keyboard alpha f [x2y]
	Character	Matrix lower left		⌞		Character ⌞ (Fontcode : 23A3 ₁₆)
	Character	Matrix lower right		⌟		Character ⌟ (Fontcode : 23A6 ₁₆)
	Character	Matrix middle left		⌠		Character ⌠ (Fontcode : 23A2 ₁₆)
	Character	Matrix middle right		⌡		Character ⌡ (Fontcode : 23A5 ₁₆)
	Character	Matrix upper left		⌢		Character ⌢ (Fontcode : 23A1 ₁₆)
	Character	Matrix upper right		⌣		Character ⌣ (Fontcode : 23A4 ₁₆)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	Measured angle		∠		Character ∠ (Fontcode : 2221 ₁₆)
	Character	Middle dot		·		Character · (Fontcode : 00B7 ₁₆)
	Character	Minus		-		Minus (Shortcut : alpha g[-] ; Info : Fontcode : 002D ₁₆)
	Character	Minus-plus		±		Character ± (Fontcode : 2213 ₁₆)
	Character	mu		μ		Character μ (Fontcode : 00B5 ₁₆)
	Character	MU		Μ		Character Μ (Fontcode : 039C ₁₆ ; Info : press [▼] for μ at [α...ω.1] gF1)
	Character	mu		μ		Character μ (Fontcode : 03BC ₁₆ ; Info : press [▲] for Μ at [Α...Ω.1] gF1)
	Character	Much greater than		»		Character » (Fontcode : 226B ₁₆)
	Character	Much less than		«		Character « (Fontcode : 226A ₁₆)
	Character	N		N		Character N (Fontcode : 004E ₁₆) ; Keyboard alpha [+/-]
	Character	n		n		Character n (Fontcode : 006E ₁₆) ; Keyboard alpha f[+/-]
	Character	N accute		Ñ		Character Ñ (Fontcode : 0143 ₁₆ ; Info : press [▼] for ñ at [aintl.3] F2)
	Character	n accute		ñ		Character ñ (Fontcode : 0144 ₁₆ ; Info : press [▲] for Ñ at [aINTL.3] F2)
	Character	N caron		Ň		Character Ň (Fontcode : 0147 ₁₆ ; Info : press [▼] for ň at [aintl.3] F3)
	Character	n caron		ň		Character ň (Fontcode : 0148 ₁₆ ; Info : press [▲] for Ň at [aINTL.3] F3)
	Character	N tilde		Ñ		Character Ñ (Fontcode : 00D1 ₁₆ ; Info : press [▼] for ñ at [aintl.3] F1)
	Character	n tilde		ñ		Character ñ (Fontcode : 00F1 ₁₆ ; Info : press [▲] for Ñ at [aINTL.3] F1)
	Character	N-ary product		∏		Character ∏ (Fontcode : 220F ₁₆)
	Character	N-ary summation		∑		Character ∑ (Fontcode : 2211 ₁₆)
	Character	Nabla		∇		Character ∇ (Fontcode : 2207 ₁₆)
	Character	Nand		⋈		Character ⋈ (Fontcode : 228C ₁₆)
	Character	Narrow Apostrophe		'		Character ' (Fontcode : 028C ₁₆)
	Character	Natural N		N		Character N (Fontcode : 2115 ₁₆)
	Character	Negative exclamation mark		⦿		Character ⦿ (Fontcode : 242F ₁₆)
	Character	Nor		̸		Character ̸ (Fontcode : 22BD ₁₆)
	Character	Not		¬		Character ¬ (Fontcode : 00AC ₁₆)
	Character	Not a subset of		⊄		Character ⊄ (Fontcode : 2284 ₁₆)
	Character	Not an element of		∉		Character ∉ (Fontcode : 2209 ₁₆)
	Character	Not equal to		≠		Character ≠ (Fontcode : 2260 ₁₆)
	Character	Not parallel to		⧻		Character ⧻ (Fontcode : 2226 ₁₆)
	Character	NU		N		Character N (Fontcode : 039D ₁₆ ; Info : press [▼] for ν at [α...ω.1] gF2)
	Character	nu		ν		Character ν (Fontcode : 03BD ₁₆ ; Info : press [▲] for N at [Α...Ω.1] gF2)
	Character	o		o		Character o (Fontcode : 004F ₁₆) ; Keyboard alpha [E]
	Character	o		o		Character o (Fontcode : 006F ₁₆) ; Keyboard alpha f[E]
	Character	o accute		ó		Character ó (Fontcode : 00D3 ₁₆ ; Info : press [▼] for ô at [aintl.3] F5)
	Character	o accute		ô		Character ô (Fontcode : 00F3 ₁₆ ; Info : press [▲] for ó at [aINTL.3] F5)
	Character	o breve		õ		Character õ (Fontcode : 014E ₁₆ ; Info : press [▼] for ö at [aintl.3] fF5)
	Character	o breve		ö		Character ö (Fontcode : 014F ₁₆ ; Info : press [▲] for ö at [aINTL.3] fF5)
	Character	o circumflex		ô		Character ô (Fontcode : 00D4 ₁₆ ; Info : press [▼] for õ at [aintl.3] F6)
	Character	o circumflex		ö		Character ö (Fontcode : 00F4 ₁₆ ; Info : press [▲] for õ at [aINTL.3] F6)
	Character	o diaresis		ö		Character ö (Fontcode : 00D6 ₁₆ ; Info : press [▼] for ø at [aintl.3] fF2)
	Character	o diaresis		ø		Character ø (Fontcode : 00F6 ₁₆ ; Info : press [▲] for ö at [aINTL.3] fF2)
	Character	o grave		ò		Character ò (Fontcode : 00D2 ₁₆ ; Info : press [▼] for ô at [aintl.3] F4)
	Character	o grave		ô		Character ô (Fontcode : 00F2 ₁₆ ; Info : press [▲] for ò at [aINTL.3] F4)
	Character	o macron		ō		Character ō (Fontcode : 014C ₁₆ ; Info : press [▼] for ô at [aintl.3] fF4)
	Character	o macron		ö		Character ö (Fontcode : 014D ₁₆ ; Info : press [▲] for ô at [aINTL.3] fF4)
	Character	o stroke		ø		Character ø (Fontcode : 00D8 ₁₆ ; Info : press [▼] for ø at [aintl.3] fF3)
	Character	o stroke		ø		Character ø (Fontcode : 00F8 ₁₆ ; Info : press [▲] for ø at [aINTL.3] fF3)
	Character	o tilde		õ		Character õ (Fontcode : 00D5 ₁₆ ; Info : press [▼] for ö at [aintl.3] fF1)
	Character	o tilde		ö		Character ö (Fontcode : 00F5 ₁₆ ; Info : press [▲] for õ at [aINTL.3] fF1)
	Character	Obelus		÷		Character ÷ (Shortcut : alpha g[÷] ; Info : Fontcode : 00F7 ₁₆)
	Character	Oblique 1		/		Character / (Fontcode : 2421 ₁₆)
	Character	Oblique 2		/		Character / (Fontcode : 2422 ₁₆)
	Character	Oblique 3		/		Character / (Fontcode : 2425 ₁₆)
	Character	Oblique 4		/		Character / (Fontcode : 2426 ₁₆)
	Character	OE		Œ		Character Œ (Fontcode : 0152 ₁₆ ; Info : press [▼] for œ at [aintl.3] fF6)
	Character	oe		œ		Character œ (Fontcode : 0153 ₁₆ ; Info : press [▲] for Œ at [aINTL.3] fF6)
	Character	OMEGA		Ω		Character Ω (Fontcode : 03A9 ₁₆ ; Info : press [▼] for ω at [α...ω.2] fF3)
	Character	omega		ω		Character ω (Fontcode : 03C9 ₁₆ ; Info : press [▲] for Ω at [Α...Ω.2] fF3)
	Character	omega tonos		ώ		Character ώ (Fontcode : 03CE ₁₆)
	Character	OMICRON		Ο		Character Ο (Fontcode : 039F ₁₆ ; Info : press [▼] for ο at [α...ω.1] gF4)
	Character	omicron		ο		Character ο (Fontcode : 03BF ₁₆ ; Info : press [▲] for Ο at [Α...Ω.1] gF4)
	Character	omicron tonos		ό		Character ό (Fontcode : 03CC ₁₆)
	Character	Or		∨		Character ∨ (Fontcode : 2228 ₁₆)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	Overflow Carry		ø		Character ø (Fontcode : 00A9 ₁₆)
	Character	P		P		Character P (Fontcode : 0050 ₁₆) ; Keyboard alpha [7]
	Character	p		p		Character p (Fontcode : 0070 ₁₆) ; Keyboard alpha f [7]
	Character	Pair of parentheses		()		Characters ()
	Character	Parallel to				Character (Fontcode : 2225 ₁₆)
	Character	Partial differential		∂		Character ∂ (Fontcode : 2202 ₁₆)
	Character	Percent		%		Character % (Fontcode : 0025 ₁₆)
	Character	Perpendicular		⊥		Character ⊥ (Fontcode : 22A5 ₁₆)
	Character	Pgm begin		⌘		Character ⌘ (Fontcode : 242B ₁₆)
	Character	PHI		Φ		Character Φ (Fontcode : 03A8 ₁₆ ; Info : press [▼] for φ at [α...ω.2] F6)
	Character	phi		φ		Character φ (Fontcode : 03C6 ₁₆ ; Info : press [▲] for Φ at [A...Ω.2] F6)
	Character	phi-math		ϕ		Character ϕ (Fontcode : 03D5 ₁₆)
	Character	PI		Π		Character Π (Fontcode : 03A0 ₁₆ ; Info : press [▼] for π at [α...ω.1] gF5)
	Character	Pipe ; Bar				Character (Fontcode : 007C ₁₆)
	Character	Planck		ħ		Character ħ (Fontcode : 210E ₁₆)
	Character	Planck / 2pi		ℏ		Character ℏ (Fontcode : 210F ₁₆)
	Character	Plus		+		Plus (Shortcut : alpha g[+] ; Info : Fontcode : 002B ₁₆)
	Character	Plus-minus		±		Character ± (Shortcut : alpha g[+/-] ; Info : Fontcode : 00B1 ₁₆)
	Character	Polar (Character)		⊙		Character ⊙ (Fontcode : 2299 ₁₆)
	Character	Pound		£		Character £ (Fontcode : 00A3 ₁₆)
	Character	Power symbol		⓪		Character ⓪ (Fontcode : 23FB ₁₆)
	Character	Printer		⌘		Character ⌘ (Fontcode : 2399 ₁₆)
	Character	Proportional to		∝		Character ∝ (Fontcode : 221D ₁₆)
	Character	PSI		Ψ		Character Ψ (Fontcode : 03A8 ₁₆ ; Info : press [▼] for ψ at [α...ω.2] fF2)
	Character	psi		ψ		Character ψ (Fontcode : 03C8 ₁₆ ; Info : press [▲] for Ψ at [A...Ω.2] fF2)
	Character	Q		Q		Character Q (Fontcode : 0051 ₁₆) ; Keyboard alpha [8]
	Character	q		q		Character q (Fontcode : 0071 ₁₆) ; Keyboard alpha f [8]
	Character	Question mark		?		Character ? (Fontcode : 003F ₁₆) ; Keyboard alpha [R/S]
	Character	Quotation mark		"		Character " (Fontcode : 0022 ₁₆)
	Character	R		R		Character R (Fontcode : 0052 ₁₆) ; Keyboard alpha [9]
	Character	r		r		Character r (Fontcode : 0072 ₁₆) ; Keyboard alpha f [9]
	Character	R accute		Ŕ		Character Ŕ (Fontcode : 0154 ₁₆ ; Info : press [▼] for ŕ at [aintl.3] gF1)
	Character	r accute		ŕ		Character ŕ (Fontcode : 0155 ₁₆ ; Info : press [▲] for Ŕ at [aintl.3] gF1)
	Character	R caron		Ř		Character Ř (Fontcode : 0158 ₁₆ ; Info : press [▼] for ř at [aintl.3] gF2)
	Character	r caron		ř		Character ř (Fontcode : 0159 ₁₆ ; Info : press [▲] for Ř at [aintl.3] gF2)
	Character	Ratio		:		Character : (Fontcode : 2236 ₁₆)
	Character	Rational Q		ℚ		Character ℚ (Fontcode : 211A ₁₆)
	Character	RB_OFF		o		Character o (Fontcode : 25EF ₁₆)
	Character	RB_ON		⓪		Character ⓪ (Fontcode : 29BF ₁₆)
	Character	Real R		R		Character R (Fontcode : 211D ₁₆)
	Character	RHO		Ρ		Character Ρ (Fontcode : 03A1 ₁₆ ; Info : press [▼] for ρ at [α...ω.2] F1)
	Character	rho		ρ		Character ρ (Fontcode : 03C1 ₁₆ ; Info : press [▲] for Ρ at [A...Ω.2] F1)
	Character	Right angle		⊞		Character ⊞ (Fontcode : 221F ₁₆)
	Character	Right arrow		→		Character → (Fontcode : 21E2 ₁₆)
	Character	Right block arrow		➔		Character ➔ (Fontcode : 2B62 ₁₆)
	Character	Right curly bracket		}		Character } (Fontcode : 007D ₁₆)
	Character	Right double quotation mark		”		Character ” (Fontcode : 201D ₁₆)
	Character	Right French quotation mark		»		Character » (Fontcode : 00BB ₁₆)
	Character	Right single quotation mark		'		Character ' (Fontcode : 2019 ₁₆)
	Character	Right square bracket		]		Character] (Fontcode : 005D ₁₆)
	Character	right tack		⌞		Character ⌞ (Fontcode : 22A2 ₁₆)
	Character	Ring		◦		Character ◦ (Fontcode : 2218 ₁₆)
	Character	S		S		Character S (Fontcode : 0053 ₁₆) ; Keyboard alpha [z]
	Character	s		s		Character s (Fontcode : 0073 ₁₆) ; Keyboard alpha f [z]
	Character	S accute		Ŝ		Character Ŝ (Fontcode : 015A ₁₆ ; Info : press [▼] for ŝ at [aintl.3] gF3)
	Character	s accute		ŝ		Character ŝ (Fontcode : 015B ₁₆ ; Info : press [▲] for Ŝ at [aintl.3] gF3)
	Character	S caron		Š		Character Š (Fontcode : 0160 ₁₆ ; Info : press [▼] for š at [aintl.3] gF5)
	Character	s caron		š		Character š (Fontcode : 0161 ₁₆ ; Info : press [▲] for Š at [aintl.3] gF5)
	Character	S cedilla		Ş		Character Ş (Fontcode : 015E ₁₆ ; Info : press [▼] for ş at [aintl.3] gF4)
	Character	s cedilla		ş		Character ş (Fontcode : 015F ₁₆ ; Info : press [▲] for Ş at [aintl.3] gF4)
	Character	SAMPI		ϐ		Character ϐ (Fontcode : 03EB ₁₆ ; Info : press [▼] for ϑ at [α...ω.2] fF4)
	Character	sampi		ϑ		Character ϑ (Fontcode : 03E1 ₁₆ ; Info : press [▲] for ϐ at [A...Ω.2] fF4)
	Character	Section		§		Character § (Fontcode : 00A7 ₁₆)
	Character	Semicolon		;		Character ; (Shortcut : alpha f[0] ; Info : Fontcode : 003B ₁₆)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	Short right arrow		➤		Character ➤ (Fontcode : 21C0 ₁₆)
	Character	SIGMA		Σ		Character Σ (Shortcut : alpha g[Σ+] ; Info : Fontcode : 03A3 ₁₆ ; press [▼] for σ at [α...ω.2]F2)
	Character	sigma		σ		Character σ (Fontcode : 03C3 ₁₆ ; Info : press [▲] for Σ at [Α...Ω.2]F2)
	Character	SIGMA		Σ		Character Σ (Fontcode : 03A3 ₁₆ ; Info : press [▼] for σ at [α...ω.2]F3)
	Character	Single high-reversed-9 quotation mark		'		Character ' (Fontcode : 201B ₁₆)
	Character	Single low-9 quotation mark		,		Character , (Fontcode : 201A ₁₆)
	Character	Slash		/		Character / (Shortcut : alpha g[R/S] ; Info : Fontcode : 002F ₁₆)
	Character	Space		" "		Character " " (Fontcode : 0020 ₁₆) ; Keyboard alpha [+]
	Character	Space symbol		␣		Character ␣ (Fontcode : 2423 ₁₆)
	Character	Spherical angle		∠		Character ∠ (Fontcode : 2222 ₁₆)
	Character	SST		∇		Character ∇ (Fontcode : 21CB ₁₆)
	Character	Stopwatch		⌚		Character ⌚ (Fontcode : 23F1 ₁₆)
	Character	Subscript -		-		Character - (Fontcode : 208B ₁₆)
	Character	Subscript +		+		Character + (Fontcode : 208A ₁₆)
	Character	Subscript 0		₀		Character ₀ (Fontcode : 2080 ₁₆)
	Character	Subscript 1		₁		Character ₁ (Fontcode : 2081 ₁₆)
	Character	Subscript 2		₂		Character ₂ (Fontcode : 2082 ₁₆)
	Character	Subscript 3		₃		Character ₃ (Fontcode : 2083 ₁₆)
	Character	Subscript 4		₄		Character ₄ (Fontcode : 2084 ₁₆)
	Character	Subscript 5		₅		Character ₅ (Fontcode : 2085 ₁₆)
	Character	Subscript 6		₆		Character ₆ (Fontcode : 2086 ₁₆)
	Character	Subscript 7		₇		Character ₇ (Fontcode : 2087 ₁₆)
	Character	Subscript 8		₈		Character ₈ (Fontcode : 2088 ₁₆)
	Character	Subscript 9		₉		Character ₉ (Fontcode : 2089 ₁₆)
	Character	Subscript A		ₐ		Character ₐ (Fontcode : 24D0 ₁₆)
	Character	Subscript a		ₐ		Character ₐ (Fontcode : 249C ₁₆)
	Character	Subscript alpha		ₐ		Character ₐ (Fontcode : 2296 ₁₆)
	Character	Subscript B		ₑ		Character ₑ (Fontcode : 24D1 ₁₆)
	Character	Subscript b		ₑ		Character ₑ (Fontcode : 249D ₁₆)
	Character	Subscript C		ₑ		Character ₑ (Fontcode : 24D2 ₁₆)
	Character	Subscript c		ₑ		Character ₑ (Fontcode : 249E ₁₆)
	Character	Subscript D		ₑ		Character ₑ (Fontcode : 24D3 ₁₆)
	Character	Subscript d		ₑ		Character ₑ (Fontcode : 249F ₁₆)
	Character	Subscript delta		ₑ		Character ₑ (Fontcode : 2297 ₁₆)
	Character	Subscript E		ₑ		Character ₑ (Fontcode : 24D4 ₁₆)
	Character	Subscript E		ₑ		Character ₑ (Fontcode : 2073 ₁₆)
	Character	Subscript e		ₑ		Character ₑ (Fontcode : 24A0 ₁₆)
	Character	Subscript earth		ₑ		Character ₑ (Fontcode : 2295 ₁₆)
	Character	Subscript F		ₑ		Character ₑ (Fontcode : 24D5 ₁₆)
	Character	Subscript f		ₑ		Character ₑ (Fontcode : 24A1 ₁₆)
	Character	Subscript G		ₑ		Character ₑ (Fontcode : 24D6 ₁₆)
	Character	Subscript g		ₑ		Character ₑ (Fontcode : 24A2 ₁₆)
	Character	Subscript H		ₑ		Character ₑ (Fontcode : 24D7 ₁₆)
	Character	Subscript h		ₑ		Character ₑ (Fontcode : 24A3 ₁₆)
	Character	Subscript I		ₑ		Character ₑ (Fontcode : 24D8 ₁₆)
	Character	Subscript i		ₑ		Character ₑ (Fontcode : 24A4 ₁₆)
	Character	Subscript infinity		∞		Character ∞ (Fontcode : 209F ₁₆)
	Character	Subscript J		ₑ		Character ₑ (Fontcode : 24D9 ₁₆)
	Character	Subscript j		ₑ		Character ₑ (Fontcode : 24A5 ₁₆)
	Character	Subscript K		ₑ		Character ₑ (Fontcode : 24DA ₁₆)
	Character	Subscript k		ₑ		Character ₑ (Fontcode : 24A6 ₁₆)
	Character	Subscript L		ₑ		Character ₑ (Fontcode : 24DB ₁₆)
	Character	Subscript l		ₑ		Character ₑ (Fontcode : 24A7 ₁₆)
	Character	Subscript M		ₑ		Character ₑ (Fontcode : 24DC ₁₆)
	Character	Subscript m		ₑ		Character ₑ (Fontcode : 24A8 ₁₆)
	Character	Subscript mu		ₑ		Character ₑ (Fontcode : 2298 ₁₆)
	Character	Subscript N		ₑ		Character ₑ (Fontcode : 24DD ₁₆)
	Character	Subscript n		ₑ		Character ₑ (Fontcode : 24A9 ₁₆)
	Character	Subscript O		ₑ		Character ₑ (Fontcode : 24DE ₁₆)
	Character	Subscript o		ₑ		Character ₑ (Fontcode : 24AA ₁₆)
	Character	Subscript P		ₑ		Character ₑ (Fontcode : 24DF ₁₆)
	Character	Subscript p		ₑ		Character ₑ (Fontcode : 24AB ₁₆)
	Character	Subscript pi		π		Character π (Shortcut : alpha g[▼]π (alpha g[R4]) ; Fontcode : 2C64 ₁₆)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	Subscript Q		q		Character q (Fontcode : 24E0 ₁₆)
	Character	Subscript q		q		Character q (Fontcode : 24AC ₁₆)
	Character	Subscript R		R		Character R (Fontcode : 24E1 ₁₆)
	Character	Subscript r		r		Character r (Fontcode : 24AD ₁₆)
	Character	Subscript S		s		Character s (Fontcode : 24E2 ₁₆)
	Character	Subscript s		s		Character s (Fontcode : 24AE ₁₆)
	Character	Subscript sun		@		Character @ (Fontcode : 229A ₁₆)
	Character	Subscript T		T		Character T (Fontcode : 24E3 ₁₆)
	Character	Subscript t		t		Character t (Fontcode : 24AF ₁₆)
	Character	Subscript U		u		Character u (Fontcode : 24E4 ₁₆)
	Character	Subscript u		u		Character u (Fontcode : 24B0 ₁₆)
	Character	Subscript V		v		Character v (Fontcode : 24E5 ₁₆)
	Character	Subscript v		v		Character v (Fontcode : 24B1 ₁₆)
	Character	Subscript W		w		Character w (Fontcode : 24E6 ₁₆)
	Character	Subscript w		w		Character w (Fontcode : 24B2 ₁₆)
	Character	Subscript X		x		Character x (Fontcode : 24E7 ₁₆)
	Character	Subscript x		x		Character x (Fontcode : 24B3 ₁₆)
	Character	Subscript Y		y		Character y (Fontcode : 24E8 ₁₆)
	Character	Subscript y		y		Character y (Fontcode : 24B4 ₁₆)
	Character	Subscript Z		z		Character z (Fontcode : 24E9 ₁₆)
	Character	Subscript z		z		Character z (Fontcode : 24B5 ₁₆)
	Character	Subset of		c		Character c (Fontcode : 2282 ₁₆)
	Character	Superscript -		-		Character - (Fontcode : 216B ₁₆)
	Character	Superscript +		+		Character + (Fontcode : 208F ₁₆)
	Character	Superscript +		+		Character + (Fontcode : 216A ₁₆)
	Character	Superscript 0		0		Character 0 (Fontcode : 2160 ₁₆)
	Character	Superscript 1		1		Character 1 (Fontcode : 2161 ₁₆)
	Character	Superscript 2		2		Character 2 (Fontcode : 2162 ₁₆)
	Character	Superscript 3		3		Character 3 (Fontcode : 2163 ₁₆)
	Character	Superscript 4		4		Character 4 (Fontcode : 2164 ₁₆)
	Character	Superscript 5		5		Character 5 (Fontcode : 2165 ₁₆)
	Character	Superscript 6		6		Character 6 (Fontcode : 2166 ₁₆)
	Character	Superscript 7		7		Character 7 (Fontcode : 2167 ₁₆)
	Character	Superscript 8		8		Character 8 (Fontcode : 2168 ₁₆)
	Character	Superscript 9		9		Character 9 (Fontcode : 2169 ₁₆)
	Character	Superscript A		A		Character A (Fontcode : 24B6 ₁₆)
	Character	Superscript a		a		Character a (Fontcode : 24B2 ₁₆)
	Character	Superscript B		B		Character B (Fontcode : 24B7 ₁₆)
	Character	Superscript b		b		Character b (Fontcode : 24B3 ₁₆)
	Character	Superscript C		C		Character C (Fontcode : 24B8 ₁₆)
	Character	Superscript c		c		Character c (Fontcode : 24B4 ₁₆)
	Character	Superscript D		D		Character D (Fontcode : 24B9 ₁₆)
	Character	Superscript d		d		Character d (Fontcode : 24B5 ₁₆)
	Character	Superscript E		E		Character E (Fontcode : 24BA ₁₆)
	Character	Superscript e		e		Character e (Fontcode : 24B6 ₁₆)
	Character	Superscript F		F		Character F (Fontcode : 24BB ₁₆)
	Character	Superscript f		f		Character f (Fontcode : 24B7 ₁₆)
	Character	Superscript f		f		Character f (Fontcode : 1DA0 ₁₆)
	Character	Superscript G		G		Character G (Fontcode : 24BC ₁₆)
	Character	Superscript g		g		Character g (Fontcode : 24B8 ₁₆)
	Character	Superscript g		g		Character g (Fontcode : 1D4D ₁₆)
	Character	Superscript H		H		Character H (Fontcode : 24BD ₁₆)
	Character	Superscript h		h		Character h (Fontcode : 24B9 ₁₆)
	Character	Superscript h		h		Character h (Fontcode : 02B0 ₁₆)
	Character	Superscript I		I		Character I (Fontcode : 24BE ₁₆)
	Character	Superscript i		i		Character i (Fontcode : 24BA ₁₆)
	Character	Superscript infinity		∞		Character ∞ (Fontcode : 209E ₁₆)
	Character	Superscript J		J		Character J (Fontcode : 24BF ₁₆)
	Character	Superscript j		j		Character j (Fontcode : 24BB ₁₆)
	Character	Superscript K		K		Character K (Fontcode : 24C0 ₁₆)
	Character	Superscript k		k		Character k (Fontcode : 24BC ₁₆)
	Character	Superscript L		L		Character L (Fontcode : 24C1 ₁₆)
	Character	Superscript l		l		Character l (Fontcode : 24BD ₁₆)
	Character	Superscript M		M		Character M (Fontcode : 24C2 ₁₆)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	Superscript m		m		Character m (Fontcode : 248E ₁₆)
	Character	Superscript N		N		Character N (Fontcode : 24C3 ₁₆)
	Character	Superscript n		n		Character n (Fontcode : 248F ₁₆)
	Character	Superscript O		O		Character O (Fontcode : 24C4 ₁₆)
	Character	Superscript o		o		Character o (Fontcode : 2490 ₁₆)
	Character	Superscript P		P		Character P (Fontcode : 24C5 ₁₆)
	Character	Superscript p		p		Character p (Fontcode : 2491 ₁₆)
	Character	Superscript pi		π		Character π (Shortcut : alpha g [▲] π (alpha g [R4]) ; Fontcode : 2C65 ₁₆)
	Character	Superscript pi r		πr		Character πr (Fontcode : 2C66 ₁₆)
	Character	Superscript Q		Q		Character Q (Fontcode : 24C6 ₁₆)
	Character	Superscript q		q		Character q (Fontcode : 2492 ₁₆)
	Character	Superscript R		R		Character R (Fontcode : 24C7 ₁₆)
	Character	Superscript r		r		Character r (Fontcode : 2493 ₁₆)
	Character	Superscript r		r		Character r (Fontcode : 02B3 ₁₆)
	Character	Superscript S		S		Character S (Fontcode : 24C8 ₁₆)
	Character	Superscript s		s		Character s (Fontcode : 2494 ₁₆)
	Character	Superscript T		T		Character T (Fontcode : 24C9 ₁₆)
	Character	Superscript t		t		Character t (Fontcode : 2495 ₁₆)
	Character	Superscript T		T		Character T (Fontcode : 1040 ₁₆)
	Character	Superscript U		U		Character U (Fontcode : 24CA ₁₆)
	Character	Superscript u		u		Character u (Fontcode : 2496 ₁₆)
	Character	Superscript V		V		Character V (Fontcode : 24CB ₁₆)
	Character	Superscript v		v		Character v (Fontcode : 2497 ₁₆)
	Character	Superscript W		W		Character W (Fontcode : 24CC ₁₆)
	Character	Superscript w		w		Character w (Fontcode : 2498 ₁₆)
	Character	Superscript X		X		Character X (Fontcode : 24CD ₁₆)
	Character	Superscript x		x		Character x (Fontcode : 2499 ₁₆)
	Character	Superscript x		x		Character x (Fontcode : 02E3 ₁₆)
	Character	Superscript Y		Y		Character Y (Fontcode : 24CE ₁₆)
	Character	Superscript y		y		Character y (Fontcode : 249A ₁₆)
	Character	Superscript Z		Z		Character Z (Fontcode : 24CF ₁₆)
	Character	Superscript z		z		Character z (Fontcode : 249B ₁₆)
	Character	Surface integral		$\oint$		Character $\oint$ (Fontcode : 222F ₁₆)
	Character	sz		β		Character β (Fontcode : 00DF ₁₆ ; Info : press [▼] for β at [aintl.3] gF6)
	Character	sz		β		Character β (Fontcode : 00DF ₁₆ ; Info : press [▲] for β at [aintl.3] gF6)
	Character	T		T		Character T (Fontcode : 0054 ₁₆) ; Keyboard alpha [4]
	Character	t		t		Character t (Fontcode : 0074 ₁₆) ; Keyboard alpha f [4]
	Character	t apostrophe		t		Character t (Fontcode : 0165 ₁₆)
	Character	T caron		T		Character T (Fontcode : 0164 ₁₆)
	Character	T cedilla		T		Character T (Fontcode : 0162 ₁₆ ; Info : press [▼] for t at [aintl.4] F1)
	Character	t cedilla		t		Character t (Fontcode : 0163 ₁₆ ; Info : press [▲] for T at [aintl.4] F1)
	Character	TAU		T		Character T (Fontcode : 03A4 ₁₆ ; Info : press [▼] for τ at [a...w.2] F4)
	Character	tau		t		Character t (Fontcode : 03C4 ₁₆ ; Info : press [▲] for T at [A...Ω.2] F4)
	Character	There does not exists		$\nexists$		Character $\nexists$ (Fontcode : 2204 ₁₆)
	Character	There exists		$\exists$		Character $\exists$ (Fontcode : 2203 ₁₆)
	Character	THETA		Θ		Character Θ (Fontcode : 0398 ₁₆ ; Info : press [▼] for θ at [a...w.1] fF3)
	Character	theta-math		θ		Character θ (Fontcode : 0398 ₁₆)
	Character	theta-script		θ		Character θ (Fontcode : 03D1 ₁₆ ; Info : press [▲] for Θ at [A...Ω.1] fF3)
	Character	Tilde		$\sim$		Character $\sim$ (Fontcode : 007E ₁₆)
	Character	Time		T		Character T (Fontcode : 2138 ₁₆)
	Character	Transposed		T		Character T (Fontcode : 22A4 ₁₆)
	Character	Triple integral		$\iiint$		Character $\iiint$ (Fontcode : 222D ₁₆)
	Character	U		U		Character U (Fontcode : 0055 ₁₆) ; Keyboard alpha [5]
	Character	u		u		Character u (Fontcode : 0075 ₁₆) ; Keyboard alpha f [5]
	Character	U accute		U		Character U (Fontcode : 00DA ₁₆ ; Info : press [▼] for u at [aintl.4] F4)
	Character	u accute		u		Character u (Fontcode : 00FA ₁₆ ; Info : press [▲] for U at [aintl.4] F4)
	Character	U breve		U		Character U (Fontcode : 016C ₁₆ ; Info : press [▼] for u at [aintl.4] fF3)
	Character	u breve		u		Character u (Fontcode : 016D ₁₆ ; Info : press [▲] for U at [aintl.4] fF3)
	Character	U circumflex		U		Character U (Fontcode : 00DB ₁₆ ; Info : press [▼] for u at [aintl.4] F5)
	Character	u circumflex		u		Character u (Fontcode : 00FB ₁₆ ; Info : press [▲] for U at [aintl.4] F5)
	Character	u circumflex		u		Character u (Fontcode : 03A2 ₁₆)
	Character	U diresis		U		Character U (Fontcode : 00DC ₁₆ ; Info : press [▼] for u at [aintl.4] F6)
	Character	u diresis		u		Character u (Fontcode : 00FC ₁₆ ; Info : press [▲] for U at [aintl.4] F6)
	Character	U grave		U		Character U (Fontcode : 00D9 ₁₆ ; Info : press [▼] for u at [aintl.4] F3)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	u grave		ù		Character ù (Fontcode : 00F9 ₁₆ ; Info : press [▲] for ù at [aintl.4] f3)
	Character	U macron		Ū		Character Ū (Fontcode : 016A ₁₆ ; Info : press [▼] for Ū at [aintl.4] fF2)
	Character	u macron		ū		Character ū (Fontcode : 016B ₁₆ ; Info : press [▲] for ū at [aintl.4] fF2)
	Character	u macron		ū		Character ū (Fontcode : 0380 ₁₆)
	Character	U ogonek		Ų		Character Ų (Fontcode : 0172 ₁₆ ; Info : press [▼] for Ų at [aintl.4] fF5)
	Character	u ogonek		ų		Character ų (Fontcode : 0173 ₁₆ ; Info : press [▲] for ų at [aintl.4] fF5)
	Character	U ring		Ů		Character Ů (Fontcode : 016E ₁₆ ; Info : press [▼] for Ů at [aintl.4] fF4)
	Character	u ring		ů		Character ů (Fontcode : 016F ₁₆ ; Info : press [▲] for ů at [aintl.4] fF4)
	Character	U tilde		Ũ		Character Ũ (Fontcode : 0168 ₁₆ ; Info : press [▼] for Ũ at [aintl.4] fF1)
	Character	u tilde		ũ		Character ũ (Fontcode : 0169 ₁₆ ; Info : press [▲] for ũ at [aintl.4] fF1)
	Character	UK		Ɑ		Character Ɑ (Fontcode : 242D ₁₆)
	Character	Underscore		Ɱ		Character Ɱ (Fontcode : 005F ₁₆) ; Keyboard alpha [-]
	Character	Undo		Ɐ		Character Ɐ (Fontcode : 238C ₁₆)
	Character	Union		∪		Character ∪ (Fontcode : 222A ₁₆)
	Character	Up arrow		↑		Character ↑ (Fontcode : 21E1 ₁₆)
	Character	Up block arrow		↗		Character ↗ (Fontcode : 2B61 ₁₆)
	Character	Up down arrow		↕		Character ↕ (Fontcode : 2195 ₁₆)
	Character	Up hollow arrow		↖		Character ↖ (Fontcode : 21E7 ₁₆)
	Character	UPSILON		Υ		Character Υ (Fontcode : 03A5 ₁₆ ; Info : press [▼] for υ at [a...w.2] F5)
	Character	upsilon		υ		Character υ (Fontcode : 03C5 ₁₆ ; Info : press [▲] for Υ at [A...Ω.2] F5)
	Character	UPSILON dialytika		Ỳ		Character Ỳ (Fontcode : 03AB ₁₆ ; Info : press [▼] for ỳ at [a...w.3] F4)
	Character	upsilon dialytika		ỳ		Character ỳ (Fontcode : 03CB ₁₆ ; Info : press [▲] for Ỳ at [A...Ω.3] F4)
	Character	upsilon dialytika tonos		ϋ		Character ϋ (Fontcode : 03B0 ₁₆)
	Character	upsilon tonos		ύ		Character ύ (Fontcode : 03CD ₁₆)
	Character	US		Ɱ		Character Ɱ (Fontcode : 242E ₁₆)
	Character	USB symbol		Ɱ		Character Ɱ (Fontcode : 2434 ₁₆)
	Character	User mode		Ɱ		Character Ɱ (Fontcode : 242C ₁₆)
	Character	V		V		Character V (Fontcode : 0056 ₁₆) ; Keyboard alpha [6]
	Character	v		v		Character v (Fontcode : 0076 ₁₆) ; Keyboard alpha f[6]
	Character	v circumflex		Ŵ		Character Ŵ (Fontcode : 038B ₁₆)
	Character	v macron		ṽ		Character ṽ (Fontcode : 0381 ₁₆)
	Character	Volume integral		∯		Character ∯ (Fontcode : 2230 ₁₆)
	Character	W		W		Character W (Fontcode : 0057 ₁₆) ; Keyboard alpha [x]
	Character	w		w		Character w (Fontcode : 0077 ₁₆) ; Keyboard alpha f[x]
	Character	W circumflex		Ẁ		Character Ẁ (Fontcode : 0174 ₁₆ ; Info : press [▼] for ẁ at [aintl.4] fF6)
	Character	w circumflex		ẁ		Character ẁ (Fontcode : 0175 ₁₆ ; Info : press [▲] for Ẁ at [aintl.4] fF6)
	Character	w macron		Ẃ		Character Ẃ (Fontcode : 0382 ₁₆)
	Character	Watch		⌚		Character ⌚ (Fontcode : 231A ₁₆)
	Character	Wide comma		,		Character , (Fontcode : 2788 ₁₆)
	Character	Wide dot		.		Character . (Fontcode : 22C5 ₁₆)
	Character	Wide period		.		Character . (Fontcode : 2789 ₁₆)
	Character	X		X		Character X (Fontcode : 0058 ₁₆) ; Keyboard alpha [1]
	Character	x		x		Character x (Fontcode : 0078 ₁₆) ; Keyboard alpha f[1]
	Character	x circumflex		Ẅ		Character Ẅ (Fontcode : 0379 ₁₆)
	Character	x macron		ẅ		Character ẅ (Fontcode : 0378 ₁₆)
	Character	x under root		Ẇ		Character Ẇ (Fontcode : 037F ₁₆)
	Character	XI		Ξ		Character Ξ (Fontcode : 039E ₁₆ ; Info : press [▼] for ξ at [a...w.1] gF3)
	Character	xi		ξ		Character ξ (Fontcode : 03BE ₁₆ ; Info : press [▲] for Ξ at [A...Ω.1] gF3)
	Character	Xor		⊕		Character ⊕ (Fontcode : 22BB ₁₆)
	Character	xth root		√		Character √ (Fontcode : 221C ₁₆)
	Character	Y		Y		Character Y (Fontcode : 0059 ₁₆) ; Keyboard alpha [2]
	Character	y		y		Character y (Fontcode : 0079 ₁₆) ; Keyboard alpha f[2]
	Character	Y acute		Ý		Character Ý (Fontcode : 00DD ₁₆ ; Info : press [▼] for ý at [aintl.4] gF1)
	Character	y acute		ý		Character ý (Fontcode : 00FD ₁₆ ; Info : press [▲] for Ý at [aintl.4] gF1)
	Character	Y circumflex		Ỳ		Character Ỳ (Fontcode : 0176 ₁₆ ; Info : press [▼] for ỳ at [aintl.4] gF2)
	Character	y circumflex		ỳ		Character ỳ (Fontcode : 0177 ₁₆ ; Info : press [▲] for Ỳ at [aintl.4] gF2)
	Character	Y diaeresis		ÿ		Character ÿ (Fontcode : 0178 ₁₆ ; Info : press [▼] for ŷ at [aintl.4] gF3)
	Character	y diaeresis		ÿ		Character ÿ (Fontcode : 00FF ₁₆ ; Info : press [▲] for ŷ at [aintl.4] gF3)
	Character	y macron		ȳ		Character ȳ (Fontcode : 0233 ₁₆)
	Character	y under root		Ȳ		Character Ȳ (Fontcode : 0232 ₁₆)
	Character	Yen		¥		Character ¥ (Fontcode : 00A5 ₁₆)
	Character	Z		Z		Character Z (Fontcode : 005A ₁₆) ; Keyboard alpha [3]
	Character	z		z		Character z (Fontcode : 007A ₁₆) ; Keyboard alpha f[3]
	Character	Z acute		Ž		Character Ž (Fontcode : 0179 ₁₆ ; Info : press [▼] for ž at [aintl.4] gF4)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Character	z acute		ž		Character ž (Fontcode : 017A ₁₆ ; Info : press [▲] for Ž at [aINTL.4] gF4)
	Character	z caron		ž		Character ž (Fontcode : 017E ₁₆ ; Info : press [▲] for ž at [aINTL.4] gF6)
	Character	z circumflex		ž		Character ž (Fontcode : 038D ₁₆)
	Character	Z dot		ž		Character ž (Fontcode : 017B ₁₆ ; Info : press [▼] for ž at [aintl.4] gF5)
	Character	z dot		ž		Character ž (Fontcode : 017C ₁₆ ; Info : press [▲] for ž at [aINTL.4] gF5)
	Character	z macron		ž		Character ž (Fontcode : 0383 ₁₆)
	Character	Zcaron		ž		Character ž (Fontcode : 017D ₁₆ ; Info : press [▼] for ž at [aintl.4] gF6)
	Character	ZETA		Ž		Character Ž (Fontcode : 0396 ₁₆ ; Info : press [▼] for Ž at [α...ω.1] fF1)
	Character	zeta		ζ		Character ζ (Fontcode : 03B6 ₁₆ ; Info : press [▲] for Z at [A...Ω.1] fF1)
Cleanup	MENU (item)	Delete items		[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) (Trailing : DELITM _menu DELITM ; Info : User menus can be defined using ASSIGN) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DELITM_page_1.pdf
	MENU	Clear		[CLR]	CLR	Clear flags, programs, registers, stacks, variables and reset calculator ; Keyboard g[←] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CLR_page_1.pdf
	MENU	Delete		[DELETE]	DELETE	Commands for clearing and deleting (user) menus, variables, programs and backup files (Info : User menus can be defined using ASSIGN) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DELETE_page_1.pdf
	MENU	Resets		[RESETS]	RESETS	Reset user menus to defaults ; delete user keyboard assignments Docs: https://47calc.com/doc/C47/Menus/C47_Menu_RESETS_page_1.pdf
	Command (nonpgm)	Backspace		←		Backspace : clear input when in Numeric Input Mode ; when not in NIM, executes DROPx ; CLX depending on whether flag ←DROP is SET or CLEAR (default CLEAR) (Info : To assign ← to a key or button, use ASN (f[1]) and spell the name as 'BKSPC' ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard [←]
	Command (nonpgm)	Clear graph		CLGRF	CLGRF	Clear graph
	Command (42)	Clear LCD		CLLCD	CLLCD	Clear the whole LCD display ; this is a 42-compatibility command
	Command	Clear (program) menu		CLMENU	CLMENU	Clear the programmable menu
	Command	Clear (user) flags		CLFall	CLFALL	Clear all user flags (numbered and lettered) (YESNO : Clear all flags? ; TI : All global user flags cleared)
	Command	Clear all menus		CLMall	CLMall	Clear assignments to all user defined menus (YESNO : Clear all user menus? ; TI : All user menus cleared)
	Command	Clear all variables		CLVall	CLVall	Clear all user variables ; clear matrices HISTO, STATS, Mat A, Mat B, Mat X (YESNO : Clear all user variables? ; TI : All user variables cleared)
	Command	Clear alpha		CLα	CLα	Clear any register or variable and initialize it with an empty text string (TAM : CLα _ Tam menu)
	Command	Clear current (program) variables		CLCVAR	CLCVAR	Clear current program variables
	Command	Clear LCD (screen)		CLLCD _{xy}	CLLCD _{xy}	Clear (part of) the display, starting at X and Y (Info : Range is X in [0, 399] and Y in [0, 239] (400 x 240 display, where (0, 0) is bottom left) ; -0. is accepted as a valid negative input for line drawing)
	Command	Clear message from display		CLD	CLD	Clear message from display
	Command	Clear modes		CLRMOD	CLRMOD	Clear all menus, entry, fraction and base modes ; when used while plotting in Program Entry Mode, leaves a graph and carries on (Shortcut : longerpress [EXIT] ; NIM longer [EXIT] ; alpha longer [EXIT] ; PEM longest [EXIT] ; EIM longer [EXIT])
	Command	Clear registers		CLREGS	CLREGS	Clear registers : general purpose 00-99 and IJKMNPQRSEFGHOUVW, local .00-.98 (if allocated) ; when stack size is 8, does not clear XYZTABCD ; when stack size is 4, does not clear XYZT, but DOES clear ABCD (YESNO : Clear registers?)
	Command	Clear stack		CLSTK	CLSTK	Clear stack registers : when stack size is 8, clears XYZTABCD ; when stack size is 4, clears XYZT ; if setting iLI is ON, the stack is filled with 0 (Long integers) ; when used while plotting, CLSTK returns to the normal screen as the stack is expected to show, implying an exit from graph and other modes (Shortcut : longerpress [←])
	Command	Clear statistics		CLZ	CLZ	Clear all statistics data ; delete matrices HISTO and STATS
	Command	Clear TVM variables		CLTVM	CLTVM	Clear TVM variables
	Command	Clear X		CLX	CLX	Clear X-register and disable automatic stack lift for next entry
	Command	Delete		DELETE	EQ.DEL	Delete equation
	Command	Delete (current) program		DELP	DELP	Delete current program and remove all assignments of the global labels in the program (careful!) (TAM : DELP _ TamLbLOnly menu)
	Command	Delete all		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 (YESNO : Delete all? ; TI : All user menus deleted ; All data, programs and definitions cleared)
	Command	Delete all menus		DELMall	DELMall	Delete all user defined menus and their assignments (YESNO : Delete all user menus? ; TI : All user menus deleted)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Delete all programs		DELPall	DELPall	Delete all user created programs and remove all assignments of all global labels used in the programs (YESNO : Delete all programs? ; TI : All user programs deleted)
	Command	Delete all variables		DELvall	DELvall	Delete all user variables and their assignments ; clear matrices HISTO, STATS, Mat A, Mat B, Mat X (YESNO : Delete all user variables? ; TI : All user variables deleted)
	Command	Delete Backup		DELBkup	DELBkup	Delete configuration backup file (LOAD, SAVE) (YESNO : Delete both backup files?)
	Command	HOME menu reset		HOME.R		Reset all HOME menu user assignments (and open menu) (TI : HOME menu reset to default)
	Command	Keyboard reset		KEYS.R		Clean (delete) all key assignments , including the Normal Mode assignment (if any) to key [Σ+] (TI : Key assignments cleaned ; Info : the Normal Mode key assignment can be set in menu Σ+ KEY if the calculator layout allows it)
	Command	MyMenu reset		MyM.R		Clean (delete) all MyMenu user assignments (TI : MyMenu menu cleaned)
	Command	Myα menu reset		Myα.R		Clean (delete) all Myα menu user assignments (TI : Myα menu cleaned)
	Command	P.FN menu reset		PFN.R		Reset all P.FN user assignments (and open menu) (TI : P.FN menu reset to default)
Compatibility	MENU (42)	42-compatible operations		[42]	42	Functions and commands that are 100% compatible with the HP-42S/DM42(n) calculators (Ref : https://47calc.com/doc/Application%20Notes/AN0026_C47_R47_DL_42S_Compatibility.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_42_page_1.pdf
	Function (42, monadic)	Alpha register to X		42ATOX	42ATOX	Convert first alpha character of text string in Alpha register (default K) to character code (hexadecimal) and remove this character ; this is a HP-42S/DM42(n) calculator compatible function (Info : For empty text string, 42ATOX returns 0 only)
	Function (42, monadic)	X to Alpha register		42XTOA	42XTOA	Append character, character code (converted to character), text string or real matrix (converted to text string) to Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible function (Info : For the 42-compatible alpha functions and commands, the Alpha register (default K) is limited to 44 characters)
	Command (42)	Alpha register Integer Part		42AIP	42AIP	Append integer part of X to text string in Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command (Info : If the register or variable does not contain a text string, 42AIP will first convert it to text string ; for the 42-compatible alpha functions and commands, the Alpha register (default K) is limited to 44 characters)
	Command (42)	Alpha register length		42ALENG	42ALENG	Get length of text string in Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command
	Command (42)	Alpha register position		42POSA	42POSA	Get position (starting from 0) of substring, character or character code in X within text string in Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command (Info : For substring not found or empty text string, 42POSA returns -1)
	Command (42)	Clear Alpha register		42CLA	42CLA	Clear Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command
	Command (42)	Print Alpha register		42PRA	42PRA	Print contents of Alpha register (default K) as text ; this is a HP-42S/DM42(n) calculator compatible command (Info : If  , 42PRA writes alpha to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI  is shown if flag SBprn is SET)
	Command (42)	Recall to Alpha register		42ARCL	42ARCL	Append contents of register or variable (formatted according to the display settings) to Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command (TAM : 42ARCL __ Tam menu ; for the 42-compatible alpha functions and commands, the Alpha register (default K) is limited to 44 characters ; Error 64 -- No string in alpha register -- may be raised)
	Command (42)	Rotate Alpha register		42AROT	42AROT	Rotate Alpha register by the number of characters specified in X (positive : shift left, negative : shift right) ; this is a HP-42S/DM42(n) calculator compatible command (Info : For empty text string, 42AROT acts as NOP ; Error 64 -- No string in alpha register -- may be raised)
	Command (42)	Shift Alpha register		42ASHF	42ASHF	Shift the six leftmost characters out of the Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command (Info : For empty text string, 42ASHF acts as NOP ; Error 64 -- No string in alpha register -- may be raised)
	Command (42)	Store from Alpha register		42ASTO	42ASTO	Store first six characters of text string in Alpha register (default K) in register or variable ; this is a HP-42S/DM42(n) calculator compatible command (TAM : 42ASTO __ Tam menu ; Error 64 -- No string in alpha register -- may be raised)
	Command (42, strike)	42BIT?		42BIT?		42BIT?
	Command (42, strike)	42ROTX		42ROTX		42ROTX
	Command (42, PEM)	Append string to Alpha register		42I-	42I-	Append text string to Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command (TAM : 42I- __ TamAlpha menu (maximum 14 characters) ; Info : for the 42-compatible alpha functions and commands, the Alpha register (default K) is limited to 44 characters ; Error 64 -- No string in alpha register -- may be raised)
	Command (42, PEM)	Enter string into Alpha register		42α	42α	Enter text string into Alpha register (default K) ; this is a HP-42S/DM42(n) calculator compatible command (TAM : 42α __ TamAlpha menu (maximum 15 characters))
	Command (42, PEM)	Key (42)		42KEY		Used in Program Entry Mode to specify soft button (1-18) of the Programmable MENU, forming program step for 42KEYG → 42KEY nn GT0 ; 42KEYX → 42KEY nn XEQ (TAM : 42KEY __ TamKey menu)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (42, PEM)	Key Execute (42)		42KEYX	42KEYX	Used in Program Entry Mode to specify the program to execute when soft button (1-18) of the Programmable MENU, [▲] (19), [▼] (20) or [EXIT] (21) is pressed ; if button 1-18 is specified, 42KEYX also creates a soft button in the Programmable MENU with the string in the Alpha register (default K) as a label (TAM : 42KEYX → 42KEY __ TamKey menu ; 42KEY nn XEQ __ TamLabel menu)
	Command (42, PEM)	Key Go to (42)		42KEYG	42KEYG	Used in Program Entry Mode to specify the label to go to when soft button (1-18) of the Programmable MENU, [▲] (19), [▼] (20) or [EXIT] (21) is pressed ; if button 1-18 is specified, 42KEYG also creates a soft button in the Programmable MENU with the string in the Alpha register (default K) as a label (TAM : 42KEYG → 42KEY __ TamKey menu ; 42KEY nn GT0 __ TamLabel menu)
	Command (42, PEM)	Prompt (42)		42PRMPT	42PRMPT	Halt program execution and view the register or variable contents (text string) in the top left part of the screen, without TI (for use in Program Entry Mode) ; if the contents include the ϵ symbol, optionally display one or more TI's at the top, X, Y, Z line ; example: "top line = ϵ X = ϵ Y = ϵ Z = " (uses Alpha register (default K)) (Info : if EN is SET, 42PRMPT will print when executed from the keyboard in NORM or TRACE ; in Program Entry Mode, EN must be SET as well)
	Command (42, PEM)	Variable menu (42)		42VRMNU	42VRMNU	Create variable menu ; uses Alpha register (default K) to store variable name when MVAR-defined button (<var...>) is pressed without NIM input (TAM : 42VRMNU __ TamLbOnly menu ; Error 25 -- No MVAR found in selected program -- may be raised)
	Command (42, PEM)	View Alpha register		42AVIEW	42AVIEW	View the Alpha register (default K) contents (text string) in the top left part of the screen, without TI (for use in Program Entry Mode) ; if the contents include the ϵ symbol, optionally display one or more TI's at the top, X, Y, Z line ; example: "top line = ϵ X = ϵ Y = ϵ Z = " ; this is a HP-42S/DM42(n) calculator compatible command (Info : If ACT is SET, 42AVIEW will print when executed from the keyboard in MAN, NORM or TRACE ; in Program Entry Mode, EN must be SET as well ; If ACT is CLEAR and EN is SET, 42AVIEW halts execution (R/S))
Complex and coordinates	Setting (pgm)	Polar notation	OFF	POLAR	POLAR $\odot$	Switch on polar notation (Info : SBI L ; $\odot$ is shown if flag SBcpX is SET)
	Setting (pgm)	Rectangular notation	ON	RECT	RECT $\ominus$	Switch on rectangular notation (Info : SBI L ; $\ominus$ is shown if flag SBcpX is SET)
	Setting	Classic (HP) Rect/Polar	ON	RP _{HP}	RP _{HP} $\boxplus$	Switch on x : Re ; y : Im (RECT) ; r ; θ (POLAR) in X, Y ; not programmable - in PEM, use flag RP _{HP} (Info : RP _{HP} OFF conforms to the way the C47 deals with complex numbers)
	Function (monadic, dyadic)	To polar		→P	→POLAR	Transform rectangular to polar coordinates (dyadic, stack conventions according to flag RP _{HP} SET or CLEAR) ; transform Complex number or 2D/3D vector to polar notation (monadic) and set POLAR tag (TI : r = ; θ = (POLAR) ; Info : in Program Entry Mode, flag RP _{HP} determines whether →POL _{HP} or →POL _{CX} is stored ; for a 3D vector in X, function →POLAR (→P) cycles through →SPH, then →CYL ; vector display depends on flag 3Dxyz, for 2D polar vector : [r θ ;i] [v ₁ v ₂] ^P ; 3D spherical vector : [p θ ;i ϕ k] _M [v ₁ v ₂ v ₃] ^S ; 3D cylindrical vector : [r θ ;jk] [v ₁ v ₂ v ₃] ^S ; Keyboard g[TAN]
	Function (monadic, dyadic)	To polar (CX)		→POL _{CX}	→POL _{CX}	Transform rectangular to polar coordinates (dyadic, stack conventions according to flag RP _{HP} CLEAR) ; transform Complex number or 2D/3D vector to polar notation (monadic) and set POLAR tag (TI : r = ; θ = (POLAR) ; Info : in Program Entry Mode, →POL _{CX} is stored when →POLAR is entered and flag RP _{HP} is CLEAR ; for a 3D vector in X, function →POL _{CX} cycles through →SPH, then →CYL ; vector display depends on flag 3Dxyz, for 2D polar vector : [r θ ;i] [v ₁ v ₂] ^P ; 3D spherical vector : [p θ ;i ϕ k] _M [v ₁ v ₂ v ₃] ^S ; 3D cylindrical vector : [r θ ;jk] [v ₁ v ₂ v ₃] ^S ;
	Function (monadic, dyadic)	To polar (HP)		→POL _{HP}	→POL _{HP}	Transform rectangular to polar coordinates (dyadic, stack conventions according to flag RP _{HP} SET) ; transform Complex number or 2D/3D vector to polar notation (monadic) and set POLAR tag (TI : r = ; θ = (POLAR) ; Info : in Program Entry Mode, →POL _{HP} is stored when →POLAR is entered and flag RP _{HP} is SET ; for a 3D vector in X, function →POL _{HP} cycles through →SPH, then →CYL ; vector display depends on flag 3Dxyz, for 2D polar vector : [r θ ;i] [v ₁ v ₂] ^P ; 3D spherical vector : [p θ ;i ϕ k] _M [v ₁ v ₂ v ₃] ^S ; 3D cylindrical vector : [r θ ;jk] [v ₁ v ₂ v ₃] ^S ;
	Function (monadic, dyadic)	To rectangular		→R	→RECT	Transform polar to rectangular coordinates (dyadic, stack conventions according to flag RP _{HP} SET or CLEAR) ; transform Complex number or 2D/3D vector to rectangular notation (monadic) and set RECT tag (TI : x : Re = ; y : Im = (RECT) ; Info : in Program Entry Mode, flag RP _{HP} determines whether →RECT _{HP} or →RECT _{CX} is stored ; vector display depends on flag 3Dxyz, for 2D rectangular vector : [i] [v ₁ v ₂] ; 3D rectangular vector : [ijk] [v ₁ v ₂ v ₃] ; Keyboard g[COS]
	Function (monadic, dyadic)	To rectangular (CX)		→RECT _{CX}	→RECT _{CX}	Transform polar to rectangular coordinates (dyadic, stack conventions according to flag RP _{HP} CLEAR) ; transform Complex number or 2D/3D vector to rectangular notation (monadic) and set RECT tag (TI : x : Re = ; y : Im = (RECT) ; Info : in Program Entry Mode, →RECT _{CX} is stored when →RECT is entered and flag RP _{HP} is CLEAR ; vector display depends on flag 3Dxyz, for 2D rectangular vector : [i] [v ₁ v ₂] ; 3D rectangular vector : [ijk] [v ₁ v ₂ v ₃] ;

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, dyadic)	To rectangular (HP)		$\rightarrow\text{RECT}_{\text{HP}}$	$\rightarrow\text{RECT}_{\text{HP}}$	Transform polar to rectangular coordinates (dyadic, stack conventions according to flag RP_{HP} SET) ; transform Complex number or 2D/3D vector to rectangular notation (monadic) and set RECT tag (TI : $x : \text{Re} = ; y : \text{Im} = (\text{RECT})$; Info : in Program Entry Mode, $\rightarrow\text{RECT}_{\text{HP}}$ is stored when $\rightarrow\text{RECT}$ is entered and flag RP_{HP} is SET ; vector display depends on flag 3Dxyz, for 2D rectangular vector : $[ij] \ [v_1 \ v_2 \]$; 3D rectangular vector : $[ijk] \ [v_1 \ v_2 \ v_3 \]$)
	Flag	Classic (HP) Rect/Polar	SET	RP_{HP}		SET for $x : \text{Re} ; y : \text{Im} (\text{RECT})$; $r ; \theta (\text{POLAR})$ in X, Y ; CLEAR for $y : \text{Im} ; x : \text{Re} (\text{RECT})$; $\theta ; r (\text{POLAR})$ in X, Y (Info : RP_{HP} CLEAR conforms to the way the C47 deals with complex numbers)
	Flag	Polar notation	CLEAR	POLAR		SET for polar notation ; CLEAR for rectangular notation (Info : SBI L ; $\odot$ is shown if flag SBcpX is SET)
Complex numbers	Symbol	Imaginary unit		i		Mathematical symbol i (Info : Displayed according to flag $\text{CPX}j$)
	Setting (pgm)	Imaginary i	ON	$\text{CPX}i$	$\text{CPX}i \odot$	Switch on displaying i as the imaginary number (Info : i or j displayed in stack and on soft buttons)
	Setting (pgm)	Imaginary j	OFF	$\text{CPX}j$	$\text{CPX}j \circ$	Switch on displaying j as the imaginary number (Info : i or j displayed in stack and on soft buttons)
	Setting	Complex multiplier	OFF	CPXmul	$\text{CPXmul} \square$	Switch on displaying Complex multiplier ; not programmable - in PEM, use flag CPXmul (Info : If OFF, trailing i or j is used ; complex multiplier is set by MULT*)
	Setting	Complex results	ON	CPXRES	$\text{CPXRES} \blacksquare$	Switch on Complex results ; not programmable - in PEM, use flag CPXRES (Info : CPXRES is auto-set when entering complex input ; SBI C ; R is shown if flag SBcr is SET)
	MENU	Complex functions		[CPX]	CPX	Complex functions (Ref : https://47calc.com/doc/Application%20Notes/AN0001_C47_R47_JM_Complex_number_engineering_example.pdf) ; Keyboard g[ENTER] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CPX_page_1.pdf
	Function (monadic)	Argument (angle)		$\angle$	$\angle$	Argument (angle) of Complex number ; Keyboard g[STO]
	Function (monadic)	Complex conjugate		conj	CONJ	Complex conjugate
	Function (monadic)	Complex to Real		$\mathbb{C} \rightarrow \mathbb{R}$	$\mathbb{C} \rightarrow \mathbb{R}$	Convert Complex to Real (in POLAR, using angle tag or ADM) (TI : $\text{Im} = ; \text{Re} = (\text{RECT})$; $\theta = ; r = (\text{POLAR})$ in X, Y)
	Function (monadic)	Imaginary part		Im	Im	Imaginary part of Complex number
	Function (monadic)	Magnitude		$ x $	$ x $	Magnitude (absolute value) of Complex number ; Keyboard f[STO]
	Function (monadic)	Real part		Re	Re	Real part of Complex number
	Function (monadic, legacy)	Absolute		$>\text{ABS}<$		Legacy version of function $ x $: Magnitude (absolute value) of Complex number (Info : Function $>\text{ABS}<$ is superceded by $ x $ on f[STO])
	Function (dyadic)	Complex number		COMPLEX	COMPLEX	Convert to or from Complex number (2.0-2.5) ; Keyboard f[ENTER]
	Function (dyadic)	Cross product ($\times$)		cross	CROSS	Cross product of vectors in X and Y
	Function (dyadic)	Dot product ($\cdot$)		dot	DOT	Dot product of vectors in X and Y
	Function (dyadic)	Real to Complex		$\mathbb{R} \rightarrow \mathbb{C}$	$\mathbb{R} \rightarrow \mathbb{C}$	Convert Real to Complex (in POLAR, using angle tag or ADM)
	Flag	Complex j	CLEAR	$\text{CPX}j$		SET for displaying j as the imaginary number ; CLEAR for displaying i as the imaginary number (Info : i or j displayed in stack and on soft buttons)
	Flag	Complex multiplier	CLEAR	CPXmul		SET for displaying Complex multiplier ; CLEAR for not displaying Complex multiplier (Info : If CLEAR, trailing i or j is used ; complex multiplier is set by MULT*)
	Flag	Complex results	SET	CPXRES		SET for Complex results ; CLEAR for Real results only (Info : CPXRES is auto-set when entering complex input ; SBI C ; R is shown if flag SBcr is SET)
	Command	Complex number (polar notation)		$i \odot$	$\text{op}_i \odot$	Enter complex number (polar notation) whether POLAR is set or not ; e.g. a $i \odot$ b ENTER results in a $\times$ b (according to ADM) (Info : In NIM, works like CC with POLAR set ; displayed according to flag $\text{CPX}j$ when in RECT mode)
	Command	Complex number (rectangular notation)		i	op_i	Enter complex number (rectangular notation) whether RECT is set or not ; e.g. a i b ENTER results in a + b i (Info : In NIM, works like CC with RECT set ; displayed according to flag $\text{CPX}j$ when in RECT mode) ; Keyboard g[SIN]
	Command	Compose-cut		CC	CC	Convert to or from Complex number (2.0-2.5) (Info : a CC b ENTER returns a+b i or a $\times$ b (using angle tag, regardless of POLAR, or ADM) ; CC returns Y : a, X : b ; also accepts 1x2 vectors as input)
	Command	Exchange real and imaginary part		$\text{Re} \leftrightarrow \text{Im}$	$\text{Re} \leftrightarrow \text{Im}$	Exchange real and imaginary part of Complex number
	Command	Unit vector or matrix		UNITV	UNITV	Unit vector or matrix for Real or Complex matrix or vector ; the result will have Euclidean magnitude 1 (Info : UNITV also applies to Complex numbers)
Constants	Constant	1st eccentricity squared		Se^2		Constant # 48 $\text{Se}^2 \text{sq.eccent1} = +6.6943799014 \times 10^{-3}$
	Constant	1st radiation constant		c_1		Constant # 6 $c_1 \text{c.radiatn1} = +3.741771852192758011367155555929985 \times 10^{-16}$ (Unit : z)
	Constant	2nd eccentricity squared		Se^2		Constant # 49 $\text{Se}^2 \text{sq.eccent2} = +6.73949674228 \times 10^{-3}$
	Constant	2nd radiation constant		c_2		Constant # 7 $c_2 \text{c.radiatn2} = +1.438776877503933802146671601543912 \times 10^{-2}$ (Unit : mK)
	Constant	Atomic mass constant		m_u		Constant # 31 $m_u \text{mass.atom} = +1.6605390689 \times 10^{-27}$ (Unit : kg)
	Constant	Avogadro's number		N_A		Constant # 36 $N_A \text{nr.avogadro} = +6.02214076 \times 10^{23}$ (Unit : /mol)
	Constant	Backbone exponent		ξ_B		Constant # 80 $\xi_B \text{exp.bbone} = +3.566668367128895828373073810012663 \times 10^{-1}$ -- see https://en.wikipedia.org/wiki/Silver_ratio
	Constant	Bohr radius		r_{Bohr}		Constant # 2 $r_{\text{Bohr}} \text{rad.bohr} = +5.29177210544 \times 10^{-11}$ (Unit : m)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Constant	Bohr's magneton		μ_B		Constant # 66 μ_B magn.both = $+9.274010065 \times 10^{-24}$ (Unit : J/T)
	Constant	Boltzmann constant		k_B		Constant # 21 k_B c.boltzmn = $+1.380649 \times 10^{-23}$ (Unit : J/K)
	Constant	Catalan's constant		G_C		Constant # 15 G_C c.catalan = $+9.159655941772190150546035149323841 \times 10^{-1}$
	Constant	Characteristic impedance of vacuum		Z_0		Constant # 55 Z_0 imp.ed.vac = $+3.76730313412 \times 10^2$ (Unit : Ω)
	Constant	Classical electron radius		r_e		Constant # 40 r_e rad.elec = $+2.8179403205 \times 10^{-15}$ (Unit : m)
	Constant	Compton wavelength of the electron		λ_C		Constant # 62 λ_C wavln.elec = $+2.42631023538 \times 10^{-12}$ (Unit : m)
	Constant	Compton wavelength of the neutron		λ_{Cn}		Constant # 63 λ_{Cn} wavln.neu = $+1.31959090382 \times 10^{-15}$ (Unit : m)
	Constant	Compton wavelength of the proton		λ_{Cp}		Constant # 64 λ_{Cp} wavln.prot = $+1.32140985360 \times 10^{-15}$ (Unit : m)
	Constant	Conductance quantum		G_0		Constant # 14 G_0 cond.quant = $+7.748091729863650646680823323308764 \times 10^{-5}$ (Unit : Ω^{-1})
	Constant	Days per year (Gregorian)		a_g		Constant # 1 a_g yr.gregor = $+3.652425 \times 10^2$ (Unit : d)
	Constant	Earth orbit (semi major axis)		$a_\oplus$		Constant # 4 $a_\oplus$ orb.earth = $+1.495979 \times 10^{11}$ (Unit : m)
	Constant	Electron magnetic moment		μ_e		Constant # 67 μ_e mgmom.elec = $-9.2847646917 \times 10^{-24}$ (Unit : J/T)
	Constant	Electron magnetic moment / Bohr's magneton		μ_e/μ_B		Constant # 68 μ_e/μ_B r.elec.bohr = -1.00115965218046
	Constant	Electron rest mass		m_e		Constant # 24 m_e mass.elec = $+9.1093837139 \times 10^{-31}$ (Unit : kg)
	Constant	Elementary charge		e		Constant # 8 e charge.elem = $+1.602176634 \times 10^{-19}$ (Unit : As)
	Constant	Energy equivalent of m_0		$m_0 c^2$		Constant # 32 $m_0 c^2$ energy.atom = $+1.4924180876 \times 10^{-10}$ (Unit : J)
	Constant	Euler-Mascheroni constant		γ_{EM}		Constant # 58 γ_{EM} c.eul.masc = $+5.772156649015328606065120900824024 \times 10^{-1}$
	Constant	Euler's		e		Constant # 9 e e.euler = $+2.718281828459045235360287471352662$
	Constant	Faraday constant		F		Constant # 10 F c.faraday = $+9.64853321233100184 \times 10^4$ (Unit : As/mol)
	Constant	Feigenbaum alpha		F_α		Constant # 11 F_α a.feigenbm = $+2.502907875095892822283902873218216$
	Constant	Feigenbaum delta		F_δ		Constant # 12 F_δ d.feigenbm = $+4.669201609102990671853203820466202$
	Constant	Fine-structure constant		α_F		Constant # 56 α_F c.finestruc = $+7.2973525643 \times 10^{-3}$
	Constant	Flattening factor		Sf^{-1}		Constant # 50 Sf^{-1} f.flatteng = $+2.98257223563 \times 10^2$
	Constant	Geocentric gravitational constant		$GM_\oplus$		Constant # 17 $GM_\oplus$ c.grav.geo = $+3.986004418 \times 10^{14}$ (Unit : m^3/s^2)
	Constant	Gerver constant		μ_G		Constant # 82 μ_G c.gerver = $+2.1953166887197$ -- see https://en.wikipedia.org/wiki/Moving_sofa_problem (Info : Moving sofa problem)
	Constant	Golden ratio		ϕ		Constant # 74 ϕ r.golden = $+1.618033988749894048204586834365638$ (Info : $\phi = (1 + \sqrt{5}) / 2$)
	Constant	Hyperfine transition frequency of ^{133}Cs		$\Delta\nu_{Cs}$		Constant # 60 $\Delta\nu_{Cs}$ frq.hypf.cs = $+9.19263177 \times 10^9$ (Unit : Hz)
	Constant	Josephson constant		K_J		Constant # 22 K_J c.josephsn = $+4.835978484169836324476582850545281 \times 10^{14}$ (Unit : Hz/V)
	Constant	Khinchin constant		K_0		Constant # 57 K_0 c.khinchin = $+2.685452001065306445309714835481796$
	Constant	Landé's electron g-factor		g_e		Constant # 16 g_e gfact.elec = -2.00231930436092
	Constant	Magnetic flux quantum		Φ_0		Constant # 75 Φ_0 fluxq.magn = $+2.067833848461929323081115412147497 \times 10^{-15}$ (Unit : Vs)
	Constant	Mass of the Earth		$M_\oplus$		Constant # 35 $M_\oplus$ mass.earth = $+5.9736 \times 10^{24}$ (Unit : kg)
	Constant	Mass of the Moon		M_{Moon}		Constant # 25 M_{Moon} mass.moon = $+7.349 \times 10^{22}$ (Unit : kg)
	Constant	Mass of the Sun		$M_\odot$		Constant # 34 $M_\odot$ mass.sun = $+1.9891 \times 10^{30}$ (Unit : kg)
	Constant	Mean radius of the Earth		$R_\oplus$		Constant # 45 $R_\oplus$ rad.earth = $+6.37101 \times 10^6$ (Unit : m)
	Constant	Mean radius of the Moon		R_{Moon}		Constant # 42 R_{Moon} rad.moon = $+1.73753 \times 10^6$ (Unit : m)
	Constant	Mean radius of the Sun		$R_\odot$		Constant # 44 $R_\odot$ rad.sun = $+6.96 \times 10^8$ (Unit : m)
	Constant	Minus infinity		$-\infty$		Constant # 77 $-\infty$ inf.minus = $-\infty$
	Constant	Molar gas constant		R		Constant # 39 R c.mol.gas = $+8.31446261815324$ (Unit : J/mol K)
	Constant	Moon orbit (semi major axis)		a_{Moon}		Constant # 3 a_{Moon} orb.moon = $+3.844 \times 10^8$ (Unit : m)
	Constant	Muon magnetic moment		μ_μ		Constant # 72 μ_μ mgmom.muon = $-4.4904483 \times 10^{-26}$ (Unit : J/T)
	Constant	Muon rest mass		m_μ		Constant # 33 m_μ mass.muon = $+1.883531627 \times 10^{-28}$ (Unit : kg)
	Constant	Neutron / proton rest mass		m_n/m_p		Constant # 27 m_n/m_p r.neu.prot = $+1.00137841946$
	Constant	Neutron magnetic moment		μ_n		Constant # 69 μ_n magmom.neu = $-9.6623653 \times 10^{-27}$ (Unit : J/T)
	Constant	Neutron rest mass		m_n		Constant # 26 m_n mass.neu = $+1.67492750056 \times 10^{-27}$ (Unit : kg)
	Constant	Newtonian constant of gravitation		G		Constant # 13 G c.grav.nwt = $+6.6743 \times 10^{-11}$ (Unit : $m^3/kg.s^2$)
	Constant	Nominal mean angular velocity of the Earth		$\omega_\oplus$		Constant # 76 $\omega_\oplus$ vangle.earth = $+7.292115 \times 10^{-5}$ (Unit : rad/s)
	Constant	Not a Number		NaN		Constant # 37 NaN not.a.nr = Not a number
	Constant	Nuclear magneton		μ_N		Constant # 71 μ_N magn.nucl = $+5.0507837393 \times 10^{-27}$ (Unit : J/T)
	Constant	pi		π		Constant # 84 π pi = $+3.141592653589793238462643383279503$
	Constant	Planck constant		h		Constant # 19 h c.planck = $+6.62607015 \times 10^{-34}$ (Unit : Js)
	Constant	Planck length		l_{PL}		Constant # 23 l_{PL} len.planck = $+1.616255 \times 10^{-35}$ (Unit : m)
	Constant	Planck mass		m_{PL}		Constant # 29 m_{PL} mass.planck = $+2.176434 \times 10^{-8}$ (Unit : kg)
	Constant	Planck temperature		T_P		Constant # 52 T_P temp.planck = $+1.416784 \times 10^{32}$ (Unit : K)
	Constant	Planck time		t_{PL}		Constant # 53 t_{PL} time.planck = $+5.391247 \times 10^{-44}$ (Unit : s)
	Constant	Plus infinity		$+\infty$		Constant # 78 $+\infty$ inf.plus = $+\infty$
	Constant	Proton / electron rest mass		m_p/m_e		Constant # 30 m_p/m_e r.prot.elec = $+1.83615267342 \times 10^3$
	Constant	Proton gyromagnetic ratio		γ_p		Constant # 59 γ_p r.gyro.prot = $+2.6752218708 \times 10^8$
	Constant	Proton magnetic moment		μ_p		Constant # 70 μ_p mgmom.prot = $+1.41060679545 \times 10^{-26}$ (Unit : J/T)
	Constant	Proton rest mass		m_p		Constant # 28 m_p mass.prot = $+1.67262192595 \times 10^{-27}$ (Unit : kg)
	Constant	Reduced Planck constant		$\hbar$		Constant # 20 $\hbar$ red.planck = $+1.054571817646156391262428003302281 \times 10^{-34}$ (Unit : Js)
	Constant	Rydberg constant		R_∞		Constant # 43 R_∞ c.rydberg = $+1.097373156815 \times 10^7$ (Unit : /mol)
	Constant	Semi-major axis of the Earth		$S_{a\oplus}$		Constant # 46 $S_{a\oplus}$ majax.earth = $+6.378137 \times 10^6$ (Unit : m)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Constant	Semi-minor axis of the Earth		$S_{b\oplus}$		Constant # 47 $S_{b\oplus}$ minax.earth = $+6.3567523142 \times 10^6$ (Unit : m)
	Constant	Silver ratio		δ_s		Constant # 81 δ_s r.silver = $+2.414213562373095048801688724209698$ -- see https://arxiv.org/pdf/2309.05050.pdf (Info : $\delta_s = 1 + \sqrt{2}$)
	Constant	Speed of light (vacuum)		c		Constant # 5 c lightspeed = $+2.99792458 \times 10^8$ (Unit : m/s)
	Constant	Standard atmospheric pressure		p_0		Constant # 38 p_0 press.atm = $+1.01325 \times 10^5$ (Unit : Pa)
	Constant	Standard Earth acceleration		$g_\oplus$		Constant # 18 $g_\oplus$ acc.earth = $+9.80665$ (Unit : m/s ²)
	Constant	Standard temperature		T_0		Constant # 51 T_0 temp.stand = $+2.7315 \times 10^2$ (Unit : K)
	Constant	Stefan-Boltzmann constant		σ_B		Constant # 73 σ_B c.stephbol = $+5.670374419184429453970996731889231 \times 10^{-8}$ (Unit : W/m ² K ⁴)
	Constant	tau		τ		Constant # 83 τ c.circle = $+6.283185307179586476925286766559006$
	Constant	Vacuum electric permittivity		ϵ_0		Constant # 61 ϵ_0 epermt.vac = $+8.8541878188 \times 10^{-12}$ (Unit : As/Vm)
	Constant	Vacuum magnetic permeability		μ_0		Constant # 65 μ_0 mpermb.vac = $+1.25663706127 \times 10^{-6}$ (Unit : Vs/Am)
	Constant	Volume of ideal gas		V_m		Constant # 54 V_m volume.gas = $+2.241396954501413773501110288675056 \times 10^{-2}$ (Unit : m ³ /mol)
	Constant	Von Klitzing constant		R_K		Constant # 41 R_K c.klitzing = $+2.581280745930450666004551670608744 \times 10^4$ (Unit : Ω)
	Constant	Zero		#		Constant # 79 zero = 0
	Command	Constant		CNST	CNST	Get value of one of the important scientific and technical constants (TAM : CNST __ TamNonReg menu (parameter 0-83) ; TI : <constant abbreviation> = <value> ; Ref : https://47calc.com/doc/C47/Reference/C47_Constants.pdf)
	Command	pi		π		Enter value of pi on the stack ; Keyboard f[R+]
Cooking unit conversions	MENU	Cooking unit conversions		[Chef:]	Chef:	Convert between cooking units ; this menu has two pages, one for the UK (imperial) units, the other for the US (customary) units : this is determined by flag CONVus, which is SET for US units on page 1 ; CLEAR for UK units on page 1 (default CLEAR) (Info : Flag CONVus may be SET for <to> <f> <to> unit conversion buttons ; CLEAR for <from> <f> <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Chef_page_1.pdf
Customisation	Setting (pgm)	[Σ+]DRG	OFF	→DRG		Assign DRG to [Σ+] hotkey for NORMAL mode ONLY & deactivate USER mode (Info : Shown in KEYMAP as [DRG] (inverse))
	Setting (pgm)	[Σ+]f/g	OFF	→f/g		Assign f/g to [Σ+] hotkey for NORMAL as well as USER mode ; disables longpress on [Σ+] and the use of this key as A in ASM two character search (Info : Shown in KEYMAP as [f/g] (inverse))
	Setting (pgm)	[Σ+]g	OFF	→g		Assign <g> to [Σ+] hotkey for NORMAL as well as USER mode ; disables longpress on [Σ+] and the use of this key as A in ASM two character search (Info : Shown in KEYMAP as [g] (inverse))
	Setting (pgm)	[Σ+]i _⊙	OFF	→i _⊙		Assign i _⊙ to [Σ+] hotkey for NORMAL mode ONLY & deactivate USER mode (Info : Shown in KEYMAP as [i _⊙] (inverse))
	Setting (pgm)	[Σ+]R↑	OFF	→R↑		Assign R↑ to [Σ+] hotkey for NORMAL mode ONLY & deactivate USER mode (Info : Shown in KEYMAP as [R↑] (inverse))
	Setting (pgm)	[Σ+]USER	OFF	→USER		Assign USER to [Σ+] hotkey for NORMAL mode ONLY & deactivate USER mode (Info : Shown in KEYMAP as [USER] (inverse))
	Setting (pgm)	[Σ+]x!	OFF	→x!		Assign x! to [Σ+] hotkey for NORMAL mode ONLY & deactivate USER mode (Info : Shown in KEYMAP as [x!] (inverse))
	Setting (pgm)	[Σ+]α	OFF	→α		Assign α to [Σ+] hotkey for NORMAL mode ONLY & deactivate USER mode (Info : Shown in KEYMAP as [α] (inverse) and in simulator as bright orange key)
	Setting (pgm)	[Σ+]Σ+	ON	→Σ+		Assign Σ+ to [Σ+] hotkey for NORMAL mode ONLY & deactivate USER mode (Info : Shown in KEYMAP as Σ+ (normal))
	Setting	HOME menu f/g shortcut	ON	HOMEf [Ⓙ]	HOMEf [Ⓙ]	Switch on HOME menu activated by triple and longerpress f/g-shift ; not programmable - in PEM, use flag HOMEf [Ⓙ] (Info : HOMEf [Ⓙ] and MyMf [Ⓙ] are linked (both cannot be ON at the same time ; HOMEf [Ⓙ] pertains to layouts containing key [f/g] only and also enables longerpress assignment to this key)
	Setting	HOME menu shown	OFF	HOMEb	HOMEb	Switch on HOME menu shown ; not programmable - in PEM, use flag HOMEb (Info : HOMEb and MyMb are linked (both cannot be ON at the same time ; applies when all menus are exited using EXIT)
	Setting	MyMenu f/g shortcut	OFF	MyMf [Ⓙ]	MyMf [Ⓙ]	Switch on MyMenu activated by triple and longerpress f/g-shift ; not programmable - in PEM, use flag MyMf [Ⓙ] (Info : MyMf [Ⓙ] and HOMEf [Ⓙ] are linked (both cannot be ON at the same time ; MyMf [Ⓙ] pertains to layouts containing key [f/g] only and also enables longerpress assignment to this key)
	Setting	MyMenu shown	ON	MyMb	MyMb	Switch on MyMenu shown ; not programmable - in PEM, use flag MyMb (Info : MyMb and HOMEb are linked (both cannot be ON at the same time ; applies when all menus are exited using EXIT)
	Setting	Select R47 layouts	OFF	R47		Select R47 layouts (f g, f [Ⓙ] g, f [Ⓙ] g, f [Ⓙ] g, f [Ⓙ] g) (Info : Initial layout selected is f g (R47v0))
	Setting	USER mode	OFF	USER		Switch on USER mode ; not programmable - in PEM, use flag USER (Info : To assign USER to a key or button, use ASN (f[1]) and spell the name as 'USER' ; during assignment, pressing USER (f[2]) opens MyMenu ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard f[2]

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU (item, nonpgm)	MyMenu		[MyM]	MyMenu	User menu presenting a ribbon to quickly access user selected functions (and menus) ; all buttons are user assignable ('riveted' look) ; set by RIBBONS ; reset using MyM.R ; default ribbon M.C47 (Shortcut : longpress [EXIT] ; NIM long [EXIT]) Docs: https://47calc.com/doc/C47/Menus/xxx1.pdf
	MENU	HOME		[HOME]	HOME	User menu to quickly access user selected menus and functions ; all buttons are user assignable ('riveted' look) ; initially populated for basic scitech options ; reset using HOME.R (Shortcut : triple [f/g] (because HOME ^g is SET) ; longerpress [f/g] (because HOME ^g is SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_HOME_page_1.pdf
	MENU	Keys		[KEYS]	KEYS	Layouts, ribbons and special key assignments (Ref : https://47calc.com/doc/C47/Reference/C47_Keyboards_side_by_side.pdf) ; Keyboard g [1] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_KEYS_page_1.pdf
	MENU	MyAlpha		[Myα]	Myα	User menu to quickly access user selected special characters ; all buttons are user assignable ('riveted' look) ; initially unpopulated ; reset using Myα.R (Shortcut : alpha long [EXIT] ; EIM long [EXIT]) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MyAlpha_page_1.pdf
	MENU	MyMenu		[MyMenu]	MyMenu	User menu presenting a ribbon to quickly access user selected functions (and menus) ; all buttons are user assignable ('riveted' look) ; set by RIBBONS ; reset using MyM.R ; default ribbon M.C47 (Shortcut : PEM longer [EXIT] ; longpress [EXIT] ; NIM long [EXIT] ; during assignment, pressing USER (f[2]) opens MyMenu ; Info : ribbons are documenten on page 1 : M.CPX ; 2 : M.ENG ; 3 : M.FIN ; 4 : M.SAV ; 5 : M.C47 ; 6 : M.R47 ; 7 : M.FIN+ ; 8 : M.SAV+) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MyMenu_page_1.pdf
	MENU	Ribbons		[RIBBONS]	RIBBONS	Selection of ribbons for MyM (MyMenu) (Info : Ribbons are predefined sets of buttons to assign to the primary (unshifted) row of MyM (MyMenu)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_RIBBONS_page_1.pdf
	MENU	Special [Σ+] hotkey assignments		[Σ+ KEY]	◻Σ+KEY	Special hotkey assignments to key [Σ+] in NORMAL mode Docs: https://47calc.com/doc/C47/Menus/C47_Menu_NRMKEY_page_1.pdf
	Layout	◻ ^g layout	OFF	◻ ^g		◻ ^g keyboard layout (R47v3) ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS.R) ; ◻ denotes the Black key : in the R47 layouts: R47v1 (◻ ^g ◻) and R47v3 (◻ ^g), the [◻] key is the hotkey that can be used for special key assignments in NORMAL mode, using menu ◻ KEY (TI : R47v3 L.Shift is ◻, R.Shift is ◻ ^g ; the layout is kept even on RESET)
	Layout	◻ ^g ◻ layout	OFF	◻ ^g ◻		◻ ^g ◻ keyboard layout (R47v1) ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS.R) ; ◻ denotes the Black key : in the R47 layouts: R47v1 (◻ ^g ◻) and R47v3 (◻ ^g), the [◻] key is the hotkey that can be used for special key assignments in NORMAL mode, using menu ◻ KEY (TI : R47v1 L.Shift is ◻ ^g , R.Shift is ◻ ; the layout is kept even on RESET)
	Layout	◻ ^g g layout	OFF	◻ ^g g		◻ ^g g keyboard layout (R47v2) ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS.R) (TI : R47v2 L.Shift is ◻ ^g , R.Shift is g ; the layout is kept even on RESET)
	Layout	C47 layout	ON	C47		C47 keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS.R) (TI : C47: Classic single shift (DM42/DM42n base) ; the layout is kept even on RESET)
	Layout	DM42 layout	OFF	DM42		DM42 keyboard layout ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS.R) (TI : DM42: Final compatibility layout ; this layout is supported for DM42 calculators without a bezel by menu BLUE47 ; the layout is kept even on RESET)
	Layout	f g layout	OFF	f g		f g keyboard layout (R47v0) ; selecting this layout changes base keys and deletes all keyboard assignments (KEYS.R) (TI : R47v0 L.Shift is f, R.Shift is g ; the layout is kept even on RESET)
	Key (black)	Black key		<empty>		Black key : in the R47 layouts: R47v1 (◻ ^g ◻) and R47v3 (◻ ^g), the [◻] key is the hotkey that can be used for special key assignments in NORMAL mode, using menu ◻ KEY (Info : Black key is not available in layout C47, use [Σ+] instead ; note that USER mode assignment to the black key is possible as well (if available) ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf)
	Flag	HOME menu f/g shortcut	SET	HOME ^g		SET for HOME menu activated by triple and longerpress f/g-shift ; CLEAR for HOME menu f/g shortcut inactive (Info : HOME ^g and MyM ^g are linked (both cannot be SET at the same time) ; HOME ^g pertains to layouts containing key [f/g] only and also enables longerpress assignment to this key)
	Flag	HOME menu shown	CLEAR	HOMEb		SET for HOME menu shown ; CLEAR for HOME menu not shown (Info : HOMEb and MyMb are linked (both cannot be SET at the same time) ; applies when all menus are exited using EXIT)
	Flag	MyMenu f/g shortcut	CLEAR	MyM ^g		SET for MyMenu activated by triple and longerpress f/g-shift ; CLEAR for MyMenu f/g shortcut inactive (Info : MyM ^g and HOME ^g are linked (both cannot be SET at the same time) ; MyM ^g pertains to layouts containing key [f/g] only and also enables longerpress assignment to this key)
	Flag	MyMenu shown	SET	MyMb		SET for MyMenu shown ; CLEAR for MyMenu not shown (Info : MyMb and HOMEb are linked (both cannot be SET at the same time) ; applies when all menus are exited using EXIT)
	Flag	USER mode	CLEAR	USER		SET for USER mode ; CLEAR for Normal Mode

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Copy [Σ+] assignment to USER		N→U		Copy [Σ+] NORMAL assignment to [Σ+] USER assignment and activate USER mode (Info : Does not copy →g or →f/g assignment)
	Command	Copy USER assignment to [Σ+]		U→N		Copy [Σ+] USER assignment to [Σ+] NORMAL assignment and deactivate USER mode
	Command	MyMenu C47		M.C47	M.C47	MyMenu primary F-key C47 ribbon ; leaves shifted rows intact ; documented as MyMenu p5 (TI : MyMenu primary F-key C47 ribbon)
	Command	MyMenu C47 Plus		M.C47+	M.C47+	MyMenu primary F-key C47 Plus ribbon ; leaves shifted rows intact ; documented as MyMenu p9 (TI : MyMenu primary F-key C47 Plus ribbon)
	Command	MyMenu Complex		M.CPX	M.CPX	MyMenu primary F-key Complex ribbon ; leaves shifted rows intact ; documented as MyMenu p1 (TI : MyMenu primary F-key Complex ribbon)
	Command	MyMenu Engineering		M.ENG	M.ENG	MyMenu primary F-key Engineering ribbon ; leaves shifted rows intact ; documented as MyMenu p2 (TI : MyMenu primary F-key Engineering ribbon ; Info : ribbon M.ENG is layout dependent)
	Command	MyMenu Financial		M.FIN	M.FIN	MyMenu primary F-key Financial ribbon ; leaves shifted rows intact ; documented as MyMenu p3 (TI : MyMenu primary F-key Financial ribbon)
	Command	MyMenu Financial Plus		M.FIN+	M.FIN+	MyMenu primary F-key Financial Plus ribbon ; leaves shifted rows intact ; documented as MyMenu p7 (TI : MyMenu primary F-key Financial Plus ribbon)
	Command	MyMenu R47		M.R47	M.R47	MyMenu primary F-key R47 ribbon ; leaves shifted rows intact ; documented as MyMenu p6 (TI : MyMenu primary F-key R47 ribbon)
	Command	MyMenu R47 Plus		M.R47+	M.R47+	MyMenu primary F-key R47 Plus ribbon ; leaves shifted rows intact ; documented as MyMenu p10 (TI : MyMenu primary F-key R47 Plus ribbon)
	Command	MyMenu Save/Load		M.SAV	M.SAV	MyMenu primary F-key Save/Load ribbon ; leaves shifted rows intact ; documented as MyMenu p4 (TI : MyMenu primary F-key Save/Load ribbon)
	Command	MyMenu Save/Load Plus		M.SAV+	M.SAV+	MyMenu primary F-key Save/Load Plus ribbon ; leaves shifted rows intact ; documented as MyMenu p8 (TI : MyMenu primary F-key Save/Load Plus ribbon)
Density conversions	Function (monadic, linked)	blob/in ³ to kg/m ³		blob/in ³ →kg/m ³		Convert blob per cubic inch to kilogram per cubic metre (TI : kg/m ³ ;))
	Function (monadic, linked)	blob/in ³ to lbs/in ³		blob/in ³ →lbs/in ³		Convert blob per cubic inch to pound per cubic inch (TI : lbs/in ³ ;))
	Function (monadic, linked)	kg/m ³ to blob/in ³		kg/m ³ →blob/in ³		Convert kilogram per cubic metre to blob per cubic inch (TI : blob/in ³ ; Info : single = lob/in ³ €)
	Function (monadic, linked)	kg/m ³ to lbs/in ³		kg/m ³ →lbs/in ³		Convert kilogram per cubic metre to pound per cubic inch (TI : lbs/in ³ ;))
	Function (monadic, linked)	kg/m ³ to t/mm ³		kg/m ³ →t/mm ³		Convert kilogram per cubic metre to metric tonne per cubic millimetre (TI : t/mm ³ ;))
	Function (monadic, linked)	lbs/in ³ to blob/in ³		lbs/in ³ →blob/in ³		Convert pound per cubic inch to blob per cubic inch (TI : blob/in ³ ; Info : single = lob/in ³ €)
	Function (monadic, linked)	lbs/in ³ to kg/m ³		lbs/in ³ →kg/m ³		Convert pound per cubic inch to kilogram per cubic metre (TI : kg/m ³ ;))
	Function (monadic, linked)	lbs/in ³ to t/mm ³		lbs/in ³ →t/mm ³		Convert pound per cubic inch to metric tonne per cubic millimetre (TI : t/mm ³ ;))
	Function (monadic, linked)	t/mm ³ to kg/m ³		t/mm ³ →kg/m ³		Convert metric tonne per cubic millimetre to kilogram per cubic metre (TI : kg/m ³ ;))
	Function (monadic, linked)	t/mm ³ to lbs/in ³		t/mm ³ →lbs/in ³		Convert metric tonne per cubic millimetre to pound per cubic inch (TI : lbs/in ³ ;))
Developer	Setting (group)	Return from HP-35 to C47 profile	ON	47		Return from HP-35 tribute to C47 profile (Info : The configuration is restored from register 35 where it was stored by setting the HP-35 tribute profile)
	Setting (group)	Set HP-35 tribute profile	OFF	HP35		Set HP-35 tribute profile (Info : The current configuration is stored in register 35 when setting the HP-35 tribute profile)
	Setting (group)	Set JM profile	OFF	JM		Set JM profile (Info : JM is one of the C47/R47 developers)
	Setting (group)	Set RJ profile	OFF	RJ		Set RJ profile (Info : RJ is one of the C47/R47 developers)
	MENU (hidden)	Developer menu		[DEV]		Developer menu Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DEV_page_1.pdf
Display	Setting (pgm)	12 hour clock	OFF	CLK12	CLK12 ○	Switch on 12h time display (Format : HH24:MM ; HH:MMam/pm; Info : SBI is shown if flag SBtime is SET)
	Setting (pgm)	24 hour clock	ON	CLK24	CLK24 ⊗	Switch on 24h time display (Format : HH24:MM ; HH:MMam/pm; Info : SBI is shown if flag SBtime is SET)
	Setting (pgm)	All digits notation	3	ALL	ALL ₃ ⊗	Set numeric display mode to ALL digits notation ; display all digits as far as possible with maximum nn decimal zeros (NDEC = nn = 0-19 ; DF = 0) (TAM : ALL __ TamNonReg menu ; Info : this setting applies to Real numbers and Complex numbers)
	Setting (pgm)	Apply SIG zeros	ON	SIG0	SIG0 ▣	Switch on applying SIG zeros (Info : 'Apply SIG zeros' means displaying zeros for the integer part and trailing zeros for the fractional part ; zeros are required for the strict interpretation of SIG as implemented in prior C47/C43 and WP34S ; if SIG0 is CLEAR, SIG may be understood as 'maximum digits' and displays the resulting value in a more commonly used way)
	Setting (pgm)	Chinese formatting	OFF	CHINA	S.CHN ◇	Set to Chinese regional formats (date, time, calendar, number formatting) (TI : First Gregorian day set: 1949-10-01 ; Week of Year rule set: Monday.Thursday)
	Setting (pgm)	Day.month.year	OFF	DMY	DMY ○	Switch on DD.MM.YYYY display mode (Format : DD.MM.YYYY ; Info : SBI is shown if flag SBdate is SET)
	Setting (pgm)	Default regional formatting	ON	DEFLT	S.DFLT ◇	Set to default regional formats (date, time, calendar, number formatting) (TI : First Gregorian day set: 1752-09-14 ; Week of Year rule set: Monday.Thursday)
	Setting (pgm)	Display precision	3	DSP	DSP ₃	Set display mode precision (only) ; applies to FIX, SCI, ENG, UNIT, SIG, ALL (TAM : DSP __ TamNonReg menu)
	Setting (pgm)	Engineering display large Real numbers	OFF	ENGOVR	ENGOVR ○	Switch on displaying large Real numbers in engineering notation

Category	Type	FullName	Default	Label	Catalog	Extended description
	Setting (pgm)	ENgineering notation	OFF	ENG	ENG ○	Set numeric display mode to engineering notation with nn+1 digits (NDEC = nn = 0-19 ; DF = 3) (TAM : ENG __ TamNonReg menu ; Info : this setting applies to Real numbers and Complex numbers)
	Setting (pgm)	European formatting	OFF	EUROP	S.EUR ◇	Set to European regional formats (date, time, calendar, number formatting) (TI : First Gregorian day set: 15.10.1582 ; Week of Year rule set: Monday.Thursday)
	Setting (pgm)	FIXed notation	OFF	FIX	FIX ○	Set numeric display mode to fixed notation with nn decimals (NDEC = nn = 0-19 ; DF = 1) (TAM : FIX __ TamNonReg menu ; Info : this setting applies to Real numbers and Complex numbers)
	Setting (pgm)	FP group size	3	FPGRP	FPGRP ₃	Set fractional part group size (2-9 digits) (TAM : FPGRP _ TamNonReg menu)
	Setting (pgm)	FP separator <none>	OFF	NONE	FNONE ○	Set fractional part separator to <none> (Info : Menu shows symbol ø)
	Setting (pgm)	FP separator comma	OFF	COM,	FCOM, ○	Set fractional part separator to comma
	Setting (pgm)	FP separator dot	OFF	DOT·	FDOT· ○	Set fractional part separator to dot
	Setting (pgm)	FP separator double space	OFF	WSPC_	FWSPC_ ○	Set fractional part separator to double space
	Setting (pgm)	FP separator narrow space	OFF	NSPC_	FNSPC_ ○	Set fractional part separator to narrow space
	Setting (pgm)	FP separator narrow tick	OFF	TICK'	FTICK' ○	Set fractional part separator to narrow tick
	Setting (pgm)	FP separator period	OFF	PER.	FPER. ○	Set fractional part separator to period
	Setting (pgm)	FP separator space	ON	SPC_	FSPC_ ø	Set fractional part separator to space
	Setting (pgm)	FP separator tick	OFF	WTICK'	FWTICK' ○	Set fractional part separator to tick
	Setting (pgm)	FP separator underscore	OFF	UNDR	FUNDR ○	Set fractional part separator to underscore
	Setting (pgm)	FP separator wide comma	OFF	WCOM,	FWCOM, ○	Set fractional part separator to wide comma
	Setting (pgm)	FP separator wide dot	OFF	WDOT·	FWDOT· ○	Set fractional part separator to wide dot
	Setting (pgm)	FP separator wide period	OFF	WPER.	FWPER. ○	Set fractional part separator to wide period
	Setting (pgm)	Indian formatting	OFF	INDIA	S.IND ◇	Set to Indian regional formats (date, time, calendar, number formatting) (TI : First Gregorian day set: 14.09.1752 ; Week of Year rule set: Sunday.Saturday)
	Setting (pgm)	IP first group extension	0	IPGRP1x	IPGRP1x ₀	Allow the first group to the left of the radix to contain one more digit (up to 1-9) than the number of digits specified by IPGRP1 ; a value of 0 means this setting is ignored (IPGRP1 is followed) (TAM : IPGRP1x _ TamNonReg menu ; Info : e.g. with first group size set to 3 and extension set to 2, up to a value of 2999 the digits remain grouped (without separator), but 3 000 and higher will not (separator space is assumed here))
	Setting (pgm)	IP first group size	0	IPGRP1	IPGRP1 ₀	Set number of digits of the first group to the left of the radix to be different from IPGRP ; a value of 0 means this setting is ignored (IPGRP is followed) (TAM : IPGRP1 _ TamNonReg menu)
	Setting (pgm)	IP group size	3	IPGRP	IPGRP ₃	Set integer part group size (2-9 digits) ; not for SCI or ENG notation (TAM : IPGRP _ TamNonReg menu)
	Setting (pgm)	IP separator <none>	OFF	NONE	INONE ○	Set integer part separator to <none> (Info : Menu shows symbol ø)
	Setting (pgm)	IP separator comma	OFF	COM,	ICOM, ○	Set integer part separator to comma
	Setting (pgm)	IP separator dot	OFF	DOT·	IDOT· ○	Set integer part separator to dot
	Setting (pgm)	IP separator double space	OFF	WSPC_	IWSPC_ ○	Set integer part separator to double space
	Setting (pgm)	IP separator narrow space	OFF	NSPC_	INSPC_ ○	Set integer part separator to narrow space
	Setting (pgm)	IP separator narrow tick	OFF	TICK'	ITICK' ○	Set integer part separator to narrow tick
	Setting (pgm)	IP separator period	OFF	PER.	IPER. ○	Set integer part separator to period
	Setting (pgm)	IP separator space	ON	SPC_	ISPC_ ø	Set integer part separator to space
	Setting (pgm)	IP separator tick	OFF	WTICK'	IWTICK' ○	Set integer part separator to tick
	Setting (pgm)	IP separator underscore	OFF	UNDR	IUNDR ○	Set integer part separator to underscore
	Setting (pgm)	IP separator wide comma	OFF	WCOM,	IWCOM, ○	Set integer part separator to wide comma
	Setting (pgm)	IP separator wide dot	OFF	WDOT·	IWDOT· ○	Set integer part separator to wide dot
	Setting (pgm)	IP separator wide period	OFF	WPER.	IWPER. ○	Set integer part separator to wide period
	Setting (pgm)	Japanese formatting	OFF	JAPAN	S.JPN ◇	Set to Japanese regional formats (date, time, calendar, number formatting) (TI : First Gregorian day set: 1873-01-01 ; Week of Year rule set: Sunday.Saturday)
	Setting (pgm)	Month/day/year	OFF	MDY	MDY ○	Switch on MM/DD/YYYY display mode (Format : MM/DD/YYYY ; Info : SBI is shown if flag SBdate is SET)
	Setting (pgm)	Multiplication symbol ·	OFF	MULT·	MULT· ○	Switch on displaying multiplication symbol ·
	Setting (pgm)	Multiplication symbol ×	ON	MULT×	MULT× ø	Switch on displaying multiplication symbol ×
	Setting (pgm)	Radix comma	OFF	COM,	RCOM, ○	Radix decimal comma
	Setting (pgm)	Radix dot	OFF	DOT·	RDOT· ○	Radix decimal dot
	Setting (pgm)	Radix period	ON	PER.	RPER. ø	Radix decimal period
	Setting (pgm)	Radix wide comma	OFF	WCOM,	RWCOM, ○	Radix decimal wide comma (0 :)
	Setting (pgm)	Radix wide dot	OFF	WDOT·	RWDOT· ○	Radix decimal wide dot
	Setting (pgm)	Radix wide period	OFF	WPER.	RWPER. ○	Radix decimal wide period
	Setting (pgm)	Scientific display large Real numbers	ON	SCIOVR	SCIOVR ø	Switch on displaying large Real numbers in scientific notation
	Setting (pgm)	SCientific notation	OFF	SCI	SCI ○	Set numeric display mode to scientific notation with nn decimals = nn+1 digits (NDEC = nn = 0-19 ; DF = 2) (TAM : SCI __ TamNonReg menu ; Info : this setting applies to Real numbers and Complex numbers)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Setting (pgm)	Set hide parameter	0	HIDE	HIDE ₀	Set parameter to hide small Real numbers or parts ; display ≈ 0 . for numbers with absolute values $< 10^{-n}$, where n in [12, 99] ; useful e.g. in matrices ; reset by HIDE 0 ; when menu XFENS is open, setting HIDE is ignored (TAM : HIDE _ TamNonReg menu ; Info : numbers smaller than the range specified by RNG are also displayed as ≈ 0 .)
	Setting (pgm)	SIGnificant digits notation	OFF	SIG	SIG 0	Set numeric display mode to significant digits notation with $nn+1$ digits (NDEC = $nn = 0-19$; DF = 4) ; switching over to scientific or engineering notation when number of rounded trailing zeros exceeds group size (IPGRP) (TAM : SIG _ TamNonReg menu ; Info : this setting applies to Real numbers and Complex numbers)
	Setting (pgm)	Time display format	0	TDISP	TDISP ₀	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 (TAM : TDISP _ TamNonReg menu)
	Setting (pgm)	UK formatting	OFF	UK	S.UK ◊	Set to UK regional formats (date, time, calendar, number formatting) (TI : First Gregorian day set: 14.09.1752 ; Week of Year rule set: Monday.Thursday)
	Setting (pgm)	Unit (prefix) notation	OFF	UNIT	UNIT 0	Set numeric display mode to unit (prefix) notation with $nn+1$ digits (NDEC = $nn = 0-19$; DF = 5) ; showing all or limited set of SI-prefixes depending on flag PFX.All ; if flag 1024 ⁿ is SET, binary prefixes are shown if available and long integer values are indicated by Z ; prefixes can be selected directly from menu Prefix (TAM : UNIT _ TamNonReg menu ; Info : this setting applies to Real numbers and Complex numbers)
	Setting (pgm)	USA formatting	OFF	USA	S.USA ◊	Set to USA regional formats (date, time, calendar, number formatting) (TI : First Gregorian day set: 09/14/1752 ; Week of Year rule set: Sunday.Saturday)
	Setting (pgm)	Year-month-day	ON	YMD	YMD 0	Switch on YYYY-MM-DD display mode (Format : YYYY-MM-DD ; Info : SBI is shown if flag SBdate is SET)
	Setting (pgm, old)	Set hide parameter	0	>HIDE<		Old version of the setting HIDE that used stack input (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use HIDE on [DISP.2]gF6 instead)
	Setting (cyclic)	Fixed-Scientific-Engineering	ALL	FSE	FSE _{ALL}	Display mode cycling (FIX ; SCI ; ENG ; UNIT ; SIG ; ALL) (Info : FSE _{FIX} ; FSE _{SCI} ; FSE _{ENG} ; FSE _{UNIT} ; FSE _{SIG} ; FSE _{ALL})
	Setting	Large Long integers display	ON	dLrgLI	dLrgLI 0	Switch on (large) Long integers to always display in large text ; not programmable - in PEM, use flag dLrgLI (Info : dLrgLI is overruled by dZ.0)
	Setting	Leading zeros	OFF	LEAD.0	LEAD.0 0	Switch on leading zeros (shortint bases 2, 4, 8, 16) ; not programmable - in PEM, use flag LEAD.0
	MENU (item, nonpgm)	Select FP separator		[FPART.]	FPART	Select fractional part separator ; button label showing current separator (.) ; symbol 0 for no separator (Info : CAT.MENUS FPART) Docs: https://47calc.com/doc/C47/Menus/xxx1.pdf
	MENU (item, nonpgm)	Select IP separator		[IPART.]	IPART	Select integer part separator ; button label showing current separator (.) ; symbol 0 for no separator (Info : CAT.MENUS IPART) Docs: https://47calc.com/doc/C47/Menus/xxx1.pdf
	MENU (item, nonpgm)	Select radix		[RADIX.]	RADIX	Select radix ; button label showing current radix (.) (Info : CAT.MENUS RADIX) Docs: https://47calc.com/doc/C47/Menus/xxx1.pdf
	MENU	Display settings		[DISP]	DISP	Display settings (Info : On DISP page 3, special indicators are used : ◊ : all group settings comply, ◊ : ≥ 1 group settings differ ; Ref : https://47calc.com/doc/Application%20Notes/AN0028_C47_R47_JM_Configurability_Tour.pdf) ; Keyboard f[E] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DISP_page_1.pdf
	MENU	Select FP separator		[FPART]	FPART	Select fractional part separator Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FPART_page_1.pdf
	MENU	Select IP separator		[IPART]	IPART	Select integer part separator Docs: https://47calc.com/doc/C47/Menus/C47_Menu_IPART_page_1.pdf
	MENU	Select radix		[RADIX]	RADIX	Select radix Docs: https://47calc.com/doc/C47/Menus/C47_Menu_RADIX_page_1.pdf
	Flag (system)	Day.month.year	CLEAR	DMY		SET for DD.MM.YYYY display mode ; CLEAR for DD.MM.YYYY display mode inactive (Format : DD.MM.YYYY ; Info : SBI is shown if flag SBdate is SET)
	Flag (system)	Month/day/year	CLEAR	MDY		SET for MM/DD/YYYY display mode ; CLEAR for MM/DD/YYYY display mode inactive (Format : MM/DD/YYYY ; Info : SBI is shown if flag SBdate is SET)
	Flag (system)	Year-month-day	SET	YMD		SET for YYYY-MM-DD display mode ; CLEAR for YYYY-MM-DD display mode inactive (Format : YYYY-MM-DD ; Info : SBI is shown if flag SBdate is SET)
	Flag	Apply SIG zeros	SET	SIG0		SET for applying SIG zeros ; CLEAR for not applying SIG zeros (Info : 'Apply SIG zeros' means displaying zeros for the integer part and trailing zeros for the fractional part ; zeros are required for the strict interpretation of SIG as implemented in prior C47/C43 and WP34S ; if SIG0 is CLEAR, SIG may be understood as 'maximum digits' and displays the resulting value in a more commonly used way)
	Flag	Engineering/Scientific display large Real number	CLEAR	ENG0VR		SET for displaying large Real numbers in engineering notation ; CLEAR for displaying large Real numbers in scientific notation

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag	Large Long integers display	SET	dLrgLI		SET for (large) Long integers to always display in large text; CLEAR for (large) Long integers to display in small text (Info : dLrgLI is overruled by dZ.0)
	Flag	Leading zeros	CLEAR	LEAD.0		SET for leading zeros (shortint bases 2, 4, 8, 16); CLEAR for no leading zeros (shortint bases 2, 4, 8, 16)
	Flag	Multiplication symbol	SET	MULT×		SET for displaying multiplication symbol ×; CLEAR for displaying multiplication symbol · (Info : Multiplication symbol is used in exponential and complex notation ; symbol may be displayed in annunciator(s) related to FRACT settings)
	Flag	Time display 24h	SET	TDM24		SET for 24h time display; CLEAR for 12h time display (Format : HH24:MM ; HH:MMam/pm; Info : SBI is shown if flag SBtime is SET)
	Command	Get hide parameter	0	HIDE#	HIDE#	Get parameter to hide small Real numbers or parts (Info : Set by HIDE)
	Command	Get number of digits displayed	3	NDEC#	NDEC#	Get number of digits displayed ; set by DSP or interactively from menu MLoads: (parameter for FIX ; SCI ; ENG ; UNIT ; SIG ; ALL) and by NDEC (Info : Range is [0, 19])
	Command	Set number of digits displayed	3	NDEC	NDEC ₃	Set number of digits displayed to X ; also set by DSP or interactively from menu MLoads: (parameter for FIX ; SCI ; ENG ; UNIT ; SIG ; ALL) (Info : Range is [0, 19])
Distance conversions	MENU	Length conversions		[Length:]	Length:	Convert between units of length (Info : Flag CONV _{HP} may be SET for <to> <←> <to> unit conversion buttons; CLEAR for <from> <→> <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Length_page_1.pdf
	Function (monadic, linked)	au to m		au→m		Convert astronomical unit to metre (TI : m:)
	Function (monadic, linked)	Banana to in		Banana→in		Convert Banana length to inch (TI : in:)
	Function (monadic, linked)	Banana to mm		Banana→mm		Convert Banana length to millimetre (TI : mm:)
	Function (monadic, linked)	brds to in		brds→in		Convert beardsecond to inch (TI : in:)
	Function (monadic, linked)	brds to m		brds→m		Convert beardsecond to metre (TI : m:)
	Function (monadic, linked)	chī to m		chī→m		Convert chī to metre (TI : m:)
	Function (monadic, linked)	cm to in		cm→in		Convert Centimetre to inch (TI : in:)
	Function (monadic, linked)	cùn to m		cùn→m		Convert cùn to metre (TI : m:)
	Function (monadic, linked)	fathom to m		fathom→m		Convert fathom to metre (TI : m:)
	Function (monadic, linked)	fēn to m		fēn→m		Convert fēn to metre (TI : m:)
	Function (monadic, linked)	ft to m		ft→m		Convert foot to metre (TI : m:)
	Function (monadic, linked)	fur to m		fur→m		Convert furlong to metre (TI : m:)
	Function (monadic, linked)	in to Banana		in→Banana		Convert inch to Banana length (TI : Banana:)
	Function (monadic, linked)	in to brds		in→brds		Convert inch to beardsecond (TI : brds:)
	Function (monadic, linked)	in to cm		in→cm		Convert inch to Centimetre (TI : cm:)
	Function (monadic, linked)	in to mm		in→mm		Convert inch to millimetre (TI : mm:)
	Function (monadic, linked)	km to mi		km→mi		Convert kilometre to mile (TI : mi:)
	Function (monadic, linked)	km to nmi		km→nmi		Convert kilometre to nautical mile (TI : nmi:)
	Function (monadic, linked)	lǐ to m		lǐ→m		Convert lǐ to metre (TI : m:)
	Function (monadic, linked)	ly to m		ly→m		Convert light year to metre (TI : m:)
	Function (monadic, linked)	m to au		m→au		Convert metre to astronomical unit (TI : au:)
	Function (monadic, linked)	m to brds		m→brds		Convert metre to beardsecond (TI : brds:)
	Function (monadic, linked)	m to chī		m→chī		Convert metre to chī (TI : chī:)
	Function (monadic, linked)	m to cùn		m→cùn		Convert metre to cùn (TI : cùn:)
	Function (monadic, linked)	m to fathom		m→fathom		Convert metre to fathom (TI : fathom:)
	Function (monadic, linked)	m to fēn		m→fēn		Convert metre to fēn (TI : fēn:)
	Function (monadic, linked)	m to ft		m→ft		Convert metre to foot (TI : ft:)
	Function (monadic, linked)	m to fur		m→fur		Convert metre to furlong (TI : fur:)
	Function (monadic, linked)	m to lǐ		m→lǐ		Convert metre to lǐ (TI : lǐ:)
	Function (monadic, linked)	m to ly		m→ly		Convert metre to light year (TI : ly:)
	Function (monadic, linked)	m to mi		m→mi		Convert metre to mile (TI : mi:)
	Function (monadic, linked)	m to nmi		m→nmi		Convert metre to nautical mile (TI : nmi:)
	Function (monadic, linked)	m to pc		m→pc		Convert metre to parsec (TI : pc:)
	Function (monadic, linked)	m to survey ft _{US}		m→survey ft _{US}		Convert metre to US survey foot (TI : survey ft _{US} ; Info : single = ey ft _{US} ←)
	Function (monadic, linked)	m to yd		m→yd		Convert metre to yard (TI : yd:)
	Function (monadic, linked)	m to yīn		m→yīn		Convert metre to yīn (TI : yīn:)
	Function (monadic, linked)	m to zhàng		m→zhàng		Convert metre to zhàng (TI : zhàng:)
	Function (monadic, linked)	mi to km		mi→km		Convert mile to kilometre (TI : km:)
	Function (monadic, linked)	mi to m		mi→m		Convert mile to metre (TI : m:)
	Function (monadic, linked)	mi to nmi		mi→nmi		Convert mile to nautical mile (TI : nmi:)
	Function (monadic, linked)	mm to Banana		mm→Banana		Convert millimetre to Banana length (TI : Banana:)
	Function (monadic, linked)	mm to in		mm→in		Convert millimetre to inch (TI : in:)
	Function (monadic, linked)	mm to point		mm→point		Convert millimetre to point (TI : point:)
	Function (monadic, linked)	nmi to km		nmi→km		Convert nautical mile to kilometre (TI : km:)
	Function (monadic, linked)	nmi to m		nmi→m		Convert nautical mile to metre (TI : m:)
	Function (monadic, linked)	nmi to mi		nmi→mi		Convert nautical mile to mile (TI : mi:)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, linked)	pc to m		pc→m		Convert parsec to metre (TI : m)
	Function (monadic, linked)	point to mm		point→mm		Convert point to millimetre (TI : mm)
	Function (monadic, linked)	survey ft _{US} to m		survey ft _{US} →m		Convert US survey foot to metre (TI : m)
	Function (monadic, linked)	yd to m		yd→m		Convert yard to metre (TI : m)
	Function (monadic, linked)	yīn to m		yīn→m		Convert yīn to metre (TI : m)
	Function (monadic, linked)	zhàng to m		zhàng→m		Convert zhàng to metre (TI : m)
Electrical engineering	MENU	Electrical engineering		[ELEC]	ELEC	Electrical engineering functions and custom programs (Ref : https://47calc.com/doc/Application%20Notes/AN0006_C47_R47_JM_ELEC.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ELEC_page_1.pdf
	Function (triadic)	Compose 3x1 matrix		zyx→M	zyx→M	Compose 3x1 matrix from Z, Y, X
	Function (triadic)	Convert from symmetrical components		012→abc	SYMtoA	Convert symmetrical components a0, a1, a2 in Z, Y, X to 3-phase a, b, c in Z, Y, X
	Function (triadic)	Convert to symmetrical components		abc→012	AtoSYM	Convert 3-phase a, b, c in Z, Y, X to symmetrical components a0, a1, a2 in Z, Y, X
	Function (triadic)	Convert triple RECT ↔ POLAR		3R↔3P	3R↔3P	Convert Real Z, Y, X to Complex Z, Y, X (according to flag POLAR) and vice versa (Info : Three-phase rectangular-polar conversion)
	Function (triadic)	Delta to Star (Wye)		Δ → Y	Δ → Y	Convert star connected impedances X, Y, Z to delta impedances X, Y, Z (TI : a : ; b : ; c :)
	Function (triadic)	Star (Wye) to Delta		Y → Δ	Y → Δ	Convert delta connected impedances X, Y, Z to star impedances X, Y, Z (TI : ca : ; ab : ; bc :)
	Function (triadic)	Triple I = V ÷ Z		I=V÷Z	I=V÷Z	X = R90 / R96 ; Y = R91 / R97 ; Z = R92 / R98
	Function (triadic)	Triple V = I × Z		V=I×Z	V=I×Z	X = R93 × R96 ; Y = R94 × R97 ; Z = R95 × R98
	Function (triadic)	Triple Z = V ÷ I		Z=V÷I	Z=V÷I	X = R90 / R93 ; Y = R91 / R94 ; Z = R92 / R95
	Function (monadic)	Decompose 3x1 matrix		M→zyx	M→zyx	Decompose 3x1 matrix to Z, Y, X
	Function (monadic)	X → BAL		X → BAL	X → BAL	Create balanced 3 phase quantities by pushing onto stack X × a, then X × a × a
	Command	Create 3x3 A-Matrix		[A]	op_A	Create 3x3 A-Matrix relating to Fortescue's Symmetrical Components -- see https://en.wikipedia.org/wiki/Symmetrical_components#A_matrix
	Command	Create 3x3 A(inverse)-Matrix		[A ⁻¹]	op_A ⁻¹	Create 3x3 A(inverse)-Matrix relating to Fortescue's Symmetrical Components -- see https://en.wikipedia.org/wiki/Symmetrical_components#A_matrix
	Command	Operator a		a	op_a	Insert value of 1 ∠ 120°
	Command	Operator a ²		a ²	op_a ²	Insert value of 1 ∠ 240°
	Command	Recall triple I		3RCL _{I93}	3RCL _{I93}	Push R93, R94, R95 to X, Y, Z ; used in menu ELEC and as dedicated registers for Virtual compound data types, as a "Triplet" stored in R93, R94, R95, interpreted as FMA (fused multiply-add) of R93 × R94 + R95 when used by extended accuracy functions in menu XFCNS ; split screen may show "No error" (Info : R93, mapped to R93, R94, R95)
	Command	Recall triple V		3RCLV ₉₀	3RCLV ₉₀	Push R90, R91, R92 to X, Y, Z ; used in menu ELEC and as dedicated registers for Virtual compound data types, as a "Triplet" stored in R90, R91, R92, interpreted as FMA (fused multiply-add) of R90 × R91 + R92 when used by extended accuracy functions in menu XFCNS ; split screen may show "No error" (Info : R90, mapped to R90, R91, R92)
	Command	Recall triple Z		3RCLZ ₉₆	3RCLZ ₉₆	Push R96, R97, R98 to X, Y, Z ; used in menu ELEC and as dedicated registers for Virtual compound data types, as a "Triplet" stored in R96, R97, R98, interpreted as FMA (fused multiply-add) of R96 × R97 + R98 when used by extended accuracy functions in menu XFCNS ; split screen may show "No error" (Info : R96, mapped to R96, R97, R98)
	Command	Store triple I		3STOI ₉₃	3STOI ₉₃	Copy X, Y, Z to R93, R94, R95 ; used in menu ELEC and as dedicated registers for Virtual compound data types, as a "Triplet" stored in R93, R94, R95, interpreted as FMA (fused multiply-add) of R93 × R94 + R95 when used by extended accuracy functions in menu XFCNS ; split screen may show "No error" (Info : R93, mapped to R93, R94, R95)
	Command	Store triple V		3STOV ₉₀	3STOV ₉₀	Copy X, Y, Z to R90, R91, R92 ; used in menu ELEC and as dedicated registers for Virtual compound data types, as a "Triplet" stored in R90, R91, R92, interpreted as FMA (fused multiply-add) of R90 × R91 + R92 when used by extended accuracy functions in menu XFCNS ; split screen may show "No error" (Info : R90, mapped to R90, R91, R92)
	Command	Store triple Z		3STOZ ₉₆	3STOZ ₉₆	Copy X, Y, Z to R96, R97, R98 ; used in menu ELEC and as dedicated registers for Virtual compound data types, as a "Triplet" stored in R96, R97, R98, interpreted as FMA (fused multiply-add) of R96 × R97 + R98 when used by extended accuracy functions in menu XFCNS ; split screen may show "No error" (Info : R96, mapped to R96, R97, R98)
Electrical engineering and characters	Function (dyadic)	Parallel impedance				Parallel impedance : (X × Y) / (X + Y)
Energy conversions	MENU	Energy conversions		[Energy:]	Energy:	Convert between units of energy (Info : Flag CONV _{HP} may be SET for <to> ↔ <to> unit conversion buttons; CLEAR for <from> → ← <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Energy_page_1.pdf
	Function (monadic, linked)	Btu to J		Btu→J		Convert British Thermal Unit to Joule (TI : J)
	Function (monadic, linked)	cal to J		cal→J		Convert calorie to Joule (TI : J)
	Function (monadic, linked)	erg to J		erg→J		Convert erg to Joule (TI : J)
	Function (monadic, linked)	eV to J		eV→J		Convert electronvolt to Joule (TI : J)
	Function (monadic, linked)	foe to J		foe→J		Convert foe to Joule (TI : J)
	Function (monadic, linked)	J to Btu		J→Btu		Convert Joule to British Thermal Unit (TI : Btu)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, linked)	J to cal		J→cal		Convert Joule to calorie (TI : cal:)
	Function (monadic, linked)	J to erg		J→erg		Convert Joule to erg (TI : erg:)
	Function (monadic, linked)	J to eV		J→eV		Convert Joule to electronvolt (TI : eV:)
	Function (monadic, linked)	J to foe		J→foe		Convert Joule to foe (TI : foe:)
	Function (monadic, linked)	J to Wh		J→Wh		Convert Joule to Watt-hour (TI : Wh:)
	Function (monadic, linked)	Wh to J		Wh→J		Convert Watt-hour to Joule (TI : J:)
Equations	Variable (user)	Variable ...		<var...>		Next variable(s) if any, of the expression entered in EQ.NEW (Info : After a value has been entered, the variable name is starred (<var...>*) to indicate it is selected for a subsequent operation)
	Variable (user)	Variable ...		<var...>		Next variable(s) if any, as created by MVAR (in RPN program) (Info : After a value has been entered, the variable name is starred (<var...>*) to indicate it is selected for a subsequent operation)
	Variable (user)	Variable 1		<var ₁ >		Variable 1 of the expression entered in EQ.NEW (Info : After a value has been entered, the variable name is starred (<var ₁ >*) to indicate it is selected for a subsequent operation)
	Variable (user)	Variable 1		<var ₁ >		Variable 1 as created by MVAR (in RPN program) (Info : After a value has been entered, the variable name is starred (<var ₁ >*) to indicate it is selected for a subsequent operation)
	Variable (user)	Variable 2		<var ₂ >*		Variable 2 of the expression entered in EQ.NEW (starred) (Info : The variable name is starred (<var ₂ >*) after a value has been entered, to indicate it is selected for a subsequent operation)
	Variable (user)	Variable 2		<var ₂ >*		Variable 2 as created by MVAR (in RPN program) (starred) (Info : The variable name is starred (<var ₂ >*) after a value has been entered, to indicate it is selected for a subsequent operation)
	Symbol	Variable x		x		Variable x
	MENU (item, nonpgm)	New equation		[NEW]	EQ.NEW	Open equation editor menu to create new equation (previous equation pushed) (Info : EIM = Equation Input Mode ; Ref : https://47calc.com/doc/C47/Reference/C47_Equation_editor.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_NEW_page_1.pdf
	MENU	Equation editor		[EQN]	EQN	Equation editor to enter multiple expressions (Info : EIM = Equation Input Mode ; when no expression has been entered, only button NEW is shown in this menu ; when two or more expressions have been entered, use [▲] or [▼] to select ; menu CAT [EQN] is available on keyboard f[+] in EIM ; Ref : https://47calc.com/doc/C47/Reference/C47_Equation_editor.pdf) ; Keyboard f[7] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_EQN_page_1.pdf
	MENU	First derivative menu [EQN]		[f']	f'	Numerically calculate first derivative using current expression, at value of starred variable and using step size of δ_a , as specified using the buttons in this menu (TI : $\delta=$; f'≈ ; Info : if no step size is entered, an automatic step size is determined by iteration in 16 steps, starting with 1/10 ; the step size will always be displayed as "δ") Docs: https://47calc.com/doc/C47/Menus/C47_Menu_f-prime_page_1.pdf
	MENU	Grapher [EQN]		[f Graph]	f Graph	Grapher (variables) menu for the algebraic expression entered in EQ.NEW ; 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 (Ref : https://47calc.com/doc/Application%20Notes/AN0013_C47_JM_Graphs.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_f_Graph_page_1.pdf
	MENU	Second derivative menu [EQN]		[f'']	f''	Numerically calculate second derivative using current expression, at value of starred variable and using step size of δ_a , as specified using the buttons in this menu (TI : $\delta=$; f''≈ ; Info : if no step size is entered, an automatic step size is determined by iteration in 16 steps, starting with 1/10 ; the step size will always be displayed as "δ") Docs: https://47calc.com/doc/C47/Menus/C47_Menu_f-double_page_1.pdf
	Command (submnu)	Edit equation		EDIT	EQ.EDIT	Open equation editor menu to edit equation (previous equation loaded) (Info : EIM = Equation Input Mode ; Ref : https://47calc.com/doc/C47/Reference/C47_Equation_editor.pdf)
	Command	Calculate result		Calc f	Calc f	Calculate value of the result for the expression (TI : f =)
	Command	First derivative [EQN]		Calc f'	Calc f'	Numerically calculate first derivative for (starred) variable (TI : f' =)
	Command	Second derivative [EQN]		Calc f''	Calc f''	Numerically calculate second derivative for (starred) variable (TI : f'' =)
EV economy conversions	Function (monadic, linked)	km/kWh to kWh/l ₁₀₀ km		km/kWh→kWh/l ₁₀₀ km		Convert kilometre per kiloWatt-hour to kiloWatt-hour per 100 kilometre (TI : kWh/100km ; Info : single = E/l ₁₀₀ km↔)
	Function (monadic, linked)	km/ℓ _e to kWh/l ₁₀₀ km		km/ℓ _e →kWh/l ₁₀₀ km		Convert kilometre per litre equivalent to kiloWatt-hour per 100 kilometre (TI : kWh/100km ; Info : single = E/l ₁₀₀ km↔)
	Function (monadic, linked)	kWh/l ₁₀₀ km to km/kWh		kWh/l ₁₀₀ km→km/kWh		Convert kiloWatt-hour per 100 kilometre to kilometre per kiloWatt-hour (TI : km/kWh ; Info : single = km/E↔)
	Function (monadic, linked)	kWh/l ₁₀₀ km to km/ℓ _e		kWh/l ₁₀₀ km→km/ℓ _e		Convert kiloWatt-hour per 100 kilometre to kilometre per litre equivalent (TI : km/ℓ _e)
	Function (monadic, linked)	kWh/l ₁₀₀ km to kWh/l ₁₀₀ mi		kWh/l ₁₀₀ km→kWh/l ₁₀₀ mi		Convert kiloWatt-hour per 100 kilometre to kiloWatt-hour per 100 mile (TI : kWh/100mi ; Info : single = E/l ₁₀₀ mi↔)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, linked)	kWh/l _{mi} to kWh/l _{km}		kWh/l _{mi} →kWh/l _{km}		Convert kiloWatt-hour per 100 mile to kiloWatt-hour per 100 kilometre (TI : kWh/100km; Info : single = E/l _{mi} kme)
	Function (monadic, linked)	kWh/l _{mi} to mi/kWh		kWh/l _{mi} →mi/kWh		Convert kiloWatt-hour per 100 mile to mile per kiloWatt-hour (TI : mi/kWh; Info : single = mi/Ee)
	Function (monadic, linked)	kWh/l _{mi} to mpge _{UK}		kWh/l _{mi} →mpge _{UK}		Convert kiloWatt-hour per 100 mile to mpg gasoline equivalent UK (TI : mpge _{UK})
	Function (monadic, linked)	kWh/l _{mi} to mpge _{US}		kWh/l _{mi} →mpge _{US}		Convert kiloWatt-hour per 100 mile to mpg gasoline equivalent US (TI : mpge _{US})
	Function (monadic, linked)	mi/kWh to kWh/l _{mi}		mi/kWh→kWh/l _{mi}		Convert mile per kiloWatt-hour to kiloWatt-hour per 100 mile (TI : kWh/100mi; Info : single = E/l _{mi} mi e)
	Function (monadic, linked)	mpge _{UK} to kWh/l _{mi}		mpge _{UK} →kWh/l _{mi}		Convert mpg gasoline equivalent UK to kiloWatt-hour per 100 mile (TI : kWh/100mi; Info : single = E/l _{mi} mi e)
	Function (monadic, linked)	mpge _{US} to kWh/l _{mi}		mpge _{US} →kWh/l _{mi}		Convert mpg gasoline equivalent US to kiloWatt-hour per 100 mile (TI : kWh/100mi; Info : single = E/l _{mi} mi e)
Extended accuracy	MENU	Extended functions		[XFCNS]	XFCNS	Extended accuracy functions for compounded input from three or six consecutive registers, effectively to 1000 digits for new models and simulator, but function not available on DM42 hardware ; when working from the stack, the compound result overwrites X, Y, Z, but when working from other registers, the compound result is pushed onto the stack ; Y will contain Long integer ; when menu XFCNS is open, setting HIDE is ignored (Split screen : Displaying X*Y+Z= <value> and TxA+B= <value> (or "Invalid register type/angle") when menu XFCNS is open (on new stack input, using ALL notation ; Info : extended accuracy functions are defined for (compound) real input and output only, or will result in NaN (Not a Number) or Error 31 -- Input data types do not match ; when menu XFCNS is open, if XYZ contains result in Virtual compound data type, SHOW displays X : XY+Z = <value> and ditto for T : TA+B and for the dedicated triple register storage, e.g. R90 : FMA: (fused multiply-add) ; Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf Docs: https://47calc.com/doc/C47/Menus/C47_Menu_XFCNS_page_1.pdf
	Function (monadic, XFN)	XFN : 10 to the power x		X10^X	X10^X	Extended accuracy function : 10 to the power x, for compound input from three consecutive stack registers : 10 ^{X*Y+Z} ; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign X10^X to f[LOG] (10 ^X) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI ^z is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Arc cosine		XACOS	XACOS	Extended accuracy function : Arc cosine, for compound input from three consecutive stack registers : ARCCOS(X * Y + Z) ; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XACOS to f[COS] (ACOS) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI ^z is displayed during calculation ; displays : Taylor Iter: m/1000; Dig: n/1040, where m is iteration counter, n is number of digits, shown if flag MONIT is SET ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Arc sine		XASIN	XASIN	Extended accuracy function : Arc sine, for compound input from three consecutive stack registers : ARCSIN(X * Y + Z) ; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XASIN to f[SIN] (ASIN) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI ^z is displayed during calculation ; displays : Taylor Iter: m/1000; Dig: n/1040, where m is iteration counter, n is number of digits, shown if flag MONIT is SET ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Arc tangent		XATAN	XATAN	Extended accuracy function : Arc tangent, for compound input from three consecutive stack registers : ARCTAN(X * Y + Z) ; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XATAN to f[TAN] (ATAN) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI ^z is displayed during calculation ; displays : Taylor Iter: m/1000; Dig: n/1040, where m is iteration counter, n is number of digits, shown if flag MONIT is SET ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Change sign		XCHS	XCHS	Extended accuracy function : Change sign, for compound input from three consecutive stack registers : - (X * Y + Z) ; returns compound result in X, Y, Z, where Y contains Long integer (0) and sign of X and Z is changed ; load state file STATE/xfnToC47user.s47 to assign XCHS to [CHS] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI ^z is displayed during calculation ; function not available on DM42 hardware)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, XFN)	XFN : Common logarithm		XLOG	XLOG	Extended accuracy function : Common logarithm, for compound input from three consecutive stack registers : $\text{LOG}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XLOG to [LOG] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Cosine		XCOS	XCOS	Extended accuracy function : Cosine, for compound input from three consecutive stack registers : $\text{COS}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XCOS to [COS] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; displays : Taylor Iter: m/1000; Dig: n/1040, where m is iteration counter, n is number of digits, shown if flag MONIT is SET ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : e to the power x		X e^x	X e^x	Extended accuracy function : e to the power x, for compound input from three consecutive stack registers : $e^{X \times Y + Z}$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign X e^x to f[LN] (e^x) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Natural logarithm		XLN	XLN	Extended accuracy function : Natural logarithm, for compound input from three consecutive stack registers : $\text{LN}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XLN to [LN] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Reciprocal		X1/X	X1/X	Extended accuracy function : Reciprocal, for compound input from three consecutive stack registers : $1 / (X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign X1/X to [1/x] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Round to decimal places		XRDP	XRDP	Extended accuracy function : Round real number to number of decimal places, subject to rounding mode (RMODE), for compound input from three consecutive stack registers : $\text{RDP}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) (TAM : XRDP ---- TamNonReg menu ; Range is [0, 1034] ; Info : function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Round to significant digits		XRSD	XRSD	Extended accuracy function : Round real number to number of significant digits, subject to rounding mode (RMODE), for compound input from three consecutive stack registers : $\text{RSD}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) (TAM : XRSD ---- TamNonReg menu ; Range is [0, 1034] ; Info : function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Set DEG tag or convert to DEG		X $\rightarrow$ DEG	X $\rightarrow$ DEG	Extended accuracy function : Set DEG tag or convert to DEG, for compound input from three consecutive stack registers, operates on X (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Set RAD tag or convert to RAD		X $\rightarrow$ RAD	X $\rightarrow$ RAD	Extended accuracy function : Set RAD tag or convert to RAD, for compound input from three consecutive stack registers, operates on X (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Sine		XSIN	XSIN	Extended accuracy function : Sine, for compound input from three consecutive stack registers : $\text{SIN}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XSIN to [SIN] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; displays : Taylor Iter: m/1000; Dig: n/1040, where m is iteration counter, n is number of digits, shown if flag MONIT is SET ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Square		XSQR	XSQR	Extended accuracy function : Square, for compound input from three consecutive stack registers : $(X \times Y + Z)^2$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XSQR to f[$\sqrt{x}$] (x^2) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Function (monadic, XFN)	XFN : Square root		XSQRT	XSQRT	Extended accuracy function : Square root, for compound input from three consecutive stack registers : $\sqrt{X \times Y + Z}$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XSQRT to f[$\sqrt{x}$] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI Σ is displayed during calculation ; function not available on DM42 hardware)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, XFN)	XFN : Tangent		XTAN	XTAN	Extended accuracy function : Tangent, for compound input from three consecutive stack registers : $\text{TAN}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XTAN to [TAN] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; displays : Taylor Iter: m/1000; Dig: n/1040, where m is iteration counter, n is number of digits, shown if flag MONIT is SET ; function not available on DM42 hardware)
	Function (monadic, cyclic, no	XFN : Degrees-radians-gradians		XDRG	XDRG	Extended accuracy function : Degrees-radians-gradians, for compound input from three consecutive stack registers : $\text{DRG}(X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : Add		XADD	XADD	Extended accuracy function : Add, for compound input from six consecutive stack registers : $(T \times A + B) + (X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XADD to [+] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : Divide		XDIV	XDIV	Extended accuracy function : Divide, for compound input from six consecutive stack registers : $(T \times A + B) / (X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XDIV to [:] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : Expanded inverse tangent		XATAN2	XATAN2	Extended accuracy function : Expanded inverse tangent, for compound input from six consecutive stack registers : $\text{ARCTAN}\{(T \times A + B) / (X \times Y + Z)\}$; returns compound result in X, Y, Z, where Y contains Long integer (0) (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; displays : Taylor Iter: m/1000; Dig: n/1040, where m is iteration counter, n is number of digits, shown if flag MONIT is SET ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : Modulo		XMOD	XMOD	Extended accuracy function : Modulo, for compound input from six consecutive stack registers : $\text{MOD}\{(T \times A + B), (X \times Y + Z)\}$; returns compound result in X, Y, Z, where Y contains Long integer (0) (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : Modulo (angle)		XMOD $\angle$	XMOD $\angle$	Extended accuracy function : Modulo (angle), for compound input from three consecutive stack registers : $\text{MOD}\{(X \times Y + Z), \langle \text{full circle} \rangle\}$, where $\langle \text{full circle} \rangle$ value is derived from angle tag or Angular Display Mode ; returns compound result in X, Y, Z, where Y contains Long integer (0) (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : Multiply		XMULT	XMULT	Extended accuracy function : Multiply, for compound input from six consecutive stack registers : $(T \times A + B) \times (X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XMULT to [x] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : Subtract		XSUB	XSUB	Extended accuracy function : Subtract, for compound input from six consecutive stack registers : $(T \times A + B) - (X \times Y + Z)$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XSUB to [-] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)
	Function (dyadic, XFN)	XFN : xth root		XXRTY	XXRTY	Extended accuracy function : xth root, for compound input from six consecutive stack registers : $\sqrt[x \times Y + Z]{T \times A + B}$; returns compound result in X, Y, Z, where Y contains Long integer (0) ; load state file STATE/xfnToC47user.s47 to assign XXRTY to g[R4] ($\sqrt[x]{y}$) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5) ; SBI $\frac{\pi}{2}$ is displayed during calculation ; function not available on DM42 hardware)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (dyadic, XFN)	XFN : y to the power x		XPOWER	XPOWER	Extended accuracy function : y to the power x, for compound input from six consecutive stack registers : (T × A + B) × Y ^z ; returns compound result in X, Y, Z, where Y contains Long integer (0); load state file STATE/xfnToC47user.s47 to assign XPOWER to f[1/x] (y*) in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5); SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Command (XFN)	XFN : Convert X to compound		X→XFN	X→XFN	Extended accuracy functionVirtual compound data type (0/1.0-1.5 ; 0/1.0 ; 0/1.0) : Convert X to Virtual compound data type (1.0-1.5 ; 0 ; 1.0) triplet in X, Y, Z ; load state file STATE/xfnToC47user.s47 to assign X→XFN to [Σ+] in USER mode (Info : Error 31 -- Input data types do not match -- is raised when input contains Complex number (2.0-2.5); SBI Σ is displayed during calculation ; function not available on DM42 hardware)
	Command (XFN)	XFN : pi		Xπ	Xπ	Extended accuracy command : Xπ, to push π onto the stack as a Virtual compound data type triplet with factor 1 of 1. × 10 ⁻⁹⁹⁹ in X, factor 2 of 1000 digits of π as 0/1.0 ; 0/1.0) ; Long integer in Y and term of 0. in Z ; load state file STATE/xfnToC47user.s47 to assign Xπ to f[R↓] (π) in USER mode (Info : SBI Σ is displayed during calculation ; function not available on DM42 hardware)
Financial	Variable (real)	TVM annual interest rate	0	I/a	I%/a	Reserved Real variable for the TVM annual interest rate as percentage (I/a) ; use 10 for 10%, not 0.1 ; displays as I _a or I in menu TVM when value is large (TI : I%/a = I%YR = APR =)
	Variable (real)	TVM compounding periods	12	cp/a	CPER/a	Reserved Real variable for the TVM compounding periods per annum (cp/a) ; displays as c _a or c in menu TVM when value ≠ 0 (TI : CPER/a = Compounding periods p.a. =)
	Variable (real)	TVM future value	0	FV	FV	Reserved Real variable for the TVM future value (FV) ; displays as F in menu TVM when value is large (TI : FV = Future Value =)
	Variable (real)	TVM number of payments	0	n	NPPER	Reserved Real variable for the TVM number of payments (n) (TI : NPPER = n pay periods =)
	Variable (real)	TVM payment	0	PMT	PMT	Reserved Real variable for the TVM payment (PMT) ; displays as P _m or P in menu TVM when value is large (TI : PMT = Payment =)
	Variable (real)	TVM payment period	12	pp/a	PPER/a	Reserved Real variable for the TVM payment period per annum (pp/a) ; displays as p _a or p in menu TVM when value ≠ 0 (TI : PPER/a = Pay periods p.a. =)
	Variable (real)	TVM present value	0	PV	PV	Reserved Real variable for the TVM present value (PV) ; displays as P _v or P in menu TVM when value is large (TI : PV = Present Value =)
	Setting (pgm)	HP12C standard precision	OFF	HP12C	HP12C □	Switch on HP12C standard precision of 2 decimals (Info : The HP12C standard precision means that PMT is rounded to 2 decimals before amortisation is computed, so the schedule matches a printed bank statement)
	Setting (pgm)	TVM begin payments	OFF	Begin	BeginP ○	Switch on TVM begin payments (Info : SBI END ; BEG is shown if flag SBTvm is SET and menu TVM, AMORT or CASHFL is open)
	Setting (pgm)	TVM end payments	ON	End	ENDP @	Switch on TVM end payments (Info : SBI END ; BEG is shown if flag SBTvm is SET and menu TVM, AMORT or CASHFL is open)
	MENU (strike)	TVM cash flow		[CASHFL]	CASHFL	TVM cash flow (Info : When menu CASHFL is open, [RCL]/[STO] activates menu TamRclTVM/TamStoTVM) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CASHFL_page_1.pdf
	MENU	Financial		[FIN]	FIN	Financial calculations including time value of money (TVM) and TVM cash flow (Ref : https://47calc.com/doc/C47/Reference/C47_Formulas.pdf) ; Keyboard g[8] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FIN_page_1.pdf
	MENU	Time value of money		[TVM]	TVM	Time value of money functions (Info : When menu TVM is open, [RCL]/[STO] activates menu TamRclTVM/TamStoTVM ; TVM variable values are displayed as subscripts ; symbol $\llcorner$ is used for very small values ; symbol $\lrcorner$ for very large values ; Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TVM_page_1.pdf
	MENU	TVM amortisation		[AMORT]	AMORT	TVM amortisation (Info : When menu AMORT is open, [RCL]/[STO] activates menu TamRclTVM/TamStoTVM ; Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_AMORT_page_1.pdf
	Function (monadic)	Delta percentage to mean		$\Delta\% \bar{x}_1$	$\Delta\% \bar{x}_1$	Delta percentage from $\bar{x}$ to x using statistics matrix (STATS) -- see Formula_DPCM EAN.png : https://47calc.com/doc/Images/Formula_DPCM EAN.png (TI : $\Delta\%$:)
	Function (monadic)	Effective TVM annual rate		$I \rightarrow \text{EFF}/a$	$I \rightarrow \text{EFF}/a$	Calculate Effective TVM annual rate from the TVM annual interest rate in I/a (variable I%/a) -- see Formula_ITOEFF.png : https://47calc.com/doc/Images/Formula_ITOEFF.png (TI : $\text{EFF}/a = \text{EFF}\% \text{YR} = \text{EAR} =$)
	Function (monadic)	Nominal TVM annual rate		$\text{EFF} \rightarrow I/a$	$\text{EFF} \rightarrow I/a$	Calculate Nominal TVM annual rate from the Effective TVM annual rate in X and copy value to I/a (variable I%/a) -- see Formula_EFFTOL.png : https://47calc.com/doc/Images/Formula_EFFTOL.png (TI : $I\%/a = I\% \text{YR} = \text{NAR} =$)
	Function (monadic)	Pct of sum and delta pct to mean		$\% \Sigma, \Delta\% \bar{x}_1$	$\% \Sigma, \Delta\% \bar{x}_1$	Percentage of x to Σx and delta percentage to mean using statistics matrix (STATS) (TI : $\Delta\%$: ; % :)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic)	Percentage of sum		%Σ	%Σ	Percentage of x to Σx : %Σ = 100 x / Σx -- see Formula_PCSIGMA.png : https://47calc.com/doc/Images/Formula_PCSIGMA.png (TI : % :)
	Function (dyadic)	Delta percent		Δ%	Δ%	Delta percentage from Y to X, keeping Y on stack -- see Formula_DELTAPC.png : https://47calc.com/doc/Images/Formula_DELTAPC.png (TI : Δ% :); Keyboard g[RCL]
	Function (dyadic)	Percent		%	%	X Percent of Y, keeping Y on stack -- see Formula_PC.png : https://47calc.com/doc/Images/Formula_PC.png ; Keyboard f[RCL]
	Function (dyadic)	Percentage of total		%T	%T	Percentage of total, keeping Y on stack -- see Formula_PCT.png : https://47calc.com/doc/Images/Formula_PCT.png (TI : % :)
	Function (dyadic)	Sales price having margin		%+MG	%+MG	Find sales price having margin of X percent over cost Y : %+MG = Y / (1 - X/100) -- see Formula_PCPMG.png : https://47calc.com/doc/Images/Formula_PCPMG.png (Info : Margin is percentage of sales price)
	Flag (system)	TVM end payments	SET	ENDPMT		SET for TVM end payments; CLEAR for TVM begin payments (Info : SBI END ; BEG is shown if flag SBtvm is SET)
	Flag	HP12C standard precision	CLEAR	HP12C		SET for HP12C standard precision of 2 decimals; CLEAR for full C47 accuracy (Info : The HP12C standard precision means that PMT is rounded to 2 decimals before amortisation is computed, so the schedule matches a printed bank statement)
	Command (old)	Get TVM annual interest rate		>RCL I/a<		Old version of the command to get TVM annual interest rate (percentage) (TI : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use RCL to get value of TVM annual interest rate on [TVM.1]F2 instead)
	Command (old)	Get TVM compounding periods		>RCL cp/a<		Old version of the command to get TVM compounding periods per annum (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use RCL to get value of TVM compounding periods on [TVM.1]fF6 instead)
	Command (old)	Get TVM future value		>RCL FV<		Old version of the command to get TVM future value (TI : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use RCL to get value of TVM future value on [TVM.1]F5 instead)
	Command (old)	Get TVM number of payments		>RCL n<		Old version of the command to get TVM number of payments (TI : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use RCL to get value of TVM number of payments on [TVM.1]F1 instead)
	Command (old)	Get TVM payment		>RCL PMT<		Old version of the command to get TVM payment (TI : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use RCL to get value of TVM payment on [TVM.1]F4 instead)
	Command (old)	Get TVM payment period		>RCL pp/a<		Old version of the command to get TVM payment period per annum (TI : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use RCL to get value of TVM payment period on [TVM.1]F6 instead)
	Command (old)	Get TVM present value		>RCL PV<		Old version of the command to get TVM present value (TI : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use RCL to get value of TVM present value on [TVM.1]F3 instead)
	Command	Arithmetic means, x only		$\bar{x}_1$	$\bar{x}_1$	Arithmetic means, x only (TI : $\bar{x}$ =)
	Command	Balance		BAL	BAL	Outstanding loan balance immediately after payment P2 is applied (TI : Balance remaining =)
	Command	First payment number	1	P1	P1 ₁	Set first payment number in the range (integer, 1 ≤ P1 ≤ n) (TI : From period P1:)
	Command	Last Payment number	12	P2	P2 ₁₂	Set last payment number in the range (integer, P1 ≤ P2 ≤ n) (TI : To period P2:)
	Command	Mean rate of return		%MRR	%MRR	Mean rate of return in percentage per period : %MRR = 100 x (x/y - 1) ^ (1/z) -- see Formula_PCMRR.png : https://47calc.com/doc/Images/Formula_PCMRR.png (TI : % :)
	Command	Step forward		NEXT	NXTPER	Step the window forward : P1 ← P1 + <window size>, P2 ← P2 + <window size>
	Command	Sum of interest		ΣINT	ΣINT	Total interest paid across payments P1 through P2, inclusive (TI : Σ of interest to P2 =)
	Command	Sum of principal		ΣPRN	ΣPRN	Total principal paid across payments P1 through P2, inclusive (TI : Σ of principal to P2 =)
Fixed pair conversions	Function (monadic, linked)	dB to fld		dB→fld		Convert decibel to field ratio (TI : fld:)
	Function (monadic, linked)	dB to pwr		dB→pwr		Convert decibel to power ratio (TI : pwr:)
	Function (monadic, linked)	fld to dB		fld→dB		Convert field ratio to decibel (TI : dB:)
	Function (monadic, linked)	HMS to HR		HMS→HR		Convert sexagesimal format hours to decimal hours (TI : decimal h: ; Ref : https://47calc.com/doc/Application%20Notes/AN0014_C47_JM_dms_hms_and_ms.pdf)
	Function (monadic, linked)	HR to HMS		HR→HMS		Convert decimal hours to sexagesimal format hours (TI : hh.mmss: ; Ref : https://47calc.com/doc/Application%20Notes/AN0014_C47_JM_dms_hms_and_ms.pdf)
	Function (monadic, linked)	in ⁴ to mm ⁴		in ⁴ →mm ⁴		Convert inch to the power 4 to millimetre to the power 4 (TI : mm ⁴ :)
	Function (monadic, linked)	in ⁶ to mm ⁶		in ⁶ →mm ⁶		Convert inch to the power 6 to millimetre to the power 6 (TI : mm ⁶ :)
	Function (monadic, linked)	kg/m to lbs/ft		kg/m→lbs/ft		Convert kilogram per metre to pound per foot (TI : lbs/ft:)
	Function (monadic, linked)	lbs/ft to kg/m		lbs/ft→kg/m		Convert pound per foot to kilogram per metre (TI : kg/m:)
	Function (monadic, linked)	mm ⁴ to in ⁴		mm ⁴ →in ⁴		Convert millimetre to the power 4 to inch to the power 4 (TI : in ⁴ :)
	Function (monadic, linked)	mm ⁶ to in ⁶		mm ⁶ →in ⁶		Convert millimetre to the power 6 to inch to the power 6 (TI : in ⁶ :)
	Function (monadic, linked)	pwr to dB		pwr→dB		Convert power ratio to decibel (TI : dB:)

Category	Type	FullName	Default	Label	Catalog	Extended description
Flags	MENU (item, nonpgm)	Configuration flags		[CFLG]	SYS.FL	Activate menu for setting system flags using FF (Flip flag) function (Ref : https://47calc.com/doc/Application%20Notes/AN0028_C47_R47_JM_Configurability_Tour.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CFLG_page_1.pdf
	MENU (ASM)	System flags		[SYS.FL]	SYS.FL	Presented in menu TamFlag for commands accessing system flags (Info : Type characters 1-2 to search ; If SYS.FL is accessed from CAT.MENUS, soft buttons return flag status, otherwise toggle flag status ; Ref : https://47calc.com/doc/C47/Reference/C47_Flags_named.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_SYS-FL_page_1.pdf
Force & Pressure conversions	MENU	Force & Pressure conversions		[F&p:]	F&p:	Convert between units of force and pressure (Info : Flag CONV _{HP} may be SET for <to> < > <to> unit conversion buttons; CLEAR for <from> > < <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_F-and-p_page_1.pdf
Force conversions	Function (monadic, linked)	kgf to lbf		kgf→lbf		Convert kilogram force to pound force (TI : lbf)
	Function (monadic, linked)	kgf to N		kgf→N		Convert kilogram force to Newton (TI : N)
	Function (monadic, linked)	lbf to kgf		lbf→kgf		Convert pound force to kilogram force (TI : kgf)
	Function (monadic, linked)	lbf to N		lbf→N		Convert pound force to Newton (TI : N)
	Function (monadic, linked)	N to kgf		N→kgf		Convert Newton to kilogram force (TI : kgf)
	Function (monadic, linked)	N to lbf		N→lbf		Convert Newton to pound force (TI : lbf)
Fractions	Setting (pgm)	Set max denominator for fractions [TAM]	64	DMX	DMX ₆₄	Set max denominator for fractions [TAM variant] ; maximum value of 32500 is set by button DMXmax on [TamNonRegMax.1]F2, stored as 0 and displayed as DMX _{max} (TAM : DMX ____ TamNonRegMax menu ; range is [2, 9999] or 0, indicating DMXmax = 32500 ; SBI is shown if flag SBfrac is SET)
	Setting (pgm)	Set significant digits for fractions	34	FDIGS	FDIGS ₃₄	Set the number of significant digits (1-34) used by the approximations of the fraction modes ; SBI ≈ indicates fractions can be approximated to be close enough (displayed if FDIGS < 34) (TAM : FDIGS __ TamNonReg menu ; Info : SBI [+≈] is shown if flag SBfrac is SET)
	Setting (pgm, older)	Set max denominator for fractions [TAM]	64	>DMX<		Older version of the setting DMX that used stack input (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use DMX on [DISP.1]fF4 instead or DMXX on [INFO.3]fF5)
	Setting (pgm, old)	Set max denominator for fractions [Old]	64	>DMX<		Old version of the setting DMX that could not use indirect addressing (Info : May be found in older user programs ; use DMX on [DISP.1]fF4 instead or DMXX on [INFO.3]fF5)
	Setting (cyclic, pgm, stack)	Fractions modes	OFF	a b/c		Set and cycle fractions modes : proper (mixed), improper fractions (denominator determined by DMX, DENFIX, DENANY) ; or fractional approximations including irrationals ; exit fractions mode via .d (g[LOG]) ; starts in mode last used or as set by flags PROPF, FRACT and IRFRAC ; When FRCYC is SET full cycle is available, when CLEAR, flag PROPF is excluded (used as is) and OFF state is included in cycle (Shortcut : <a> [.] [.] <c> [.] ; Info : IRFRAC and FRACT are linked (both cannot be SET at the same time) ; SBI ×1 or -1 indicates that approximate multiples or fractions can be used, of the irrational numbers $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, $\sqrt{7}$, ϕ , π^{-2} , π^{-1} , $\sqrt{\pi}$, π , π^2 , e^{-2} , e^{-1} , e and e^2 (with tolerance 10^{-24}) ; Prefix ≈ + <approximation> ; multiplication symbol SET for displaying multiplication symbol × ; SBI is shown if flag SBfrac is SET ; Ref : https://47calc.com/doc/C47/Reference/C47_Fractions.pdf)
	Setting	Denominator any	OFF	DENANY	DENANY □	Switch on any denominator (up to value set interactively from menu DISP (DMX TAM variant) and from menu [INFO.3]fF5 (DMXX)) in fraction display mode ; not programmable - in PEM, use flag DENANY (Info : DENANY and DENFIX are linked (both cannot be ON at the same time ; SBI [+×] is shown if flag SBfrac is SET)
	Setting	Denominator fixed	OFF	DENFIX	DENFIX □	Switch on fixed denominator (set interactively from menu DISP (DMX TAM variant) and from menu [INFO.3]fF5 (DMXX)) in fraction display mode ; not programmable - in PEM, use flag DENFIX (Info : DENFIX and DENANY are linked (both cannot be ON at the same time ; SBI [+f] is shown if flag SBfrac is SET)
	Setting	Fraction mode full cycle	OFF	FRCYC	FRCYC □	Switch on full cycle on key a b/c (g[+]) ; not programmable - in PEM, use flag FRCYC (Info : When FRCYC is ON full cycle is available, when OFF, flag PROPF is excluded (used as is) and OFF state is included in cycle)
	Setting	Fractions mode	OFF	FRACT	FRACT □	Switch on fractions mode ; not programmable - in PEM, use flag FRACT (Info : FRACT and IRFRAC are linked (both cannot be ON at the same time ; /n or /max denotes maximum denominator (set interactively from menu DISP (DMX TAM variant) and from menu [INFO.3]fF5 (DMXX)) ; prefixes < and > may be shown as needed for rational fractions ; also see flag IRFRAC)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Setting	Irrational fractions mode	OFF	IRFRAC	IRFRAC ☐	Switch on irrational fractions mode ; not programmable - in PEM, use flag IRFRAC (Info : IRFRAC and FRACT are linked (both cannot be ON at the same time) ; SBI $\times 1$ or $\cdot 1$ indicates that approximate multiples or fractions can be used, of the irrational numbers $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, $\sqrt{7}$, ϕ , π^{-2} , π^{-1} , $\sqrt{\pi}$, π , π^2 , e^{-2} , e^{-1} , e and e^2 (with tolerance 10^{-24}) ; Prefix $\approx$ + <approximation> ; multiplication symbol SET for displaying multiplication symbol $\times$; CLEAR for displaying multiplication symbol $\cdot$ or SET for displaying multiplication symbol $\times$; CLEAR for displaying multiplication symbol $\cdot$ according to flag MULT $\times$; SBI $\approx$ indicates FDIGS < 34; SBI is shown if flag SBfrac is SET ; exit fraction modes by .d or CLRMOD (longerpress [EXIT]))
	Setting	Proper fractions	ON	PROPFR	PROPFR ☐	Switch on proper fractions (mixed fractions) ; not programmable - in PEM, use flag PROPFR
	Function (monadic)	Fractional Part		FP	FP	Fractional Part of real number (Shortcut : #TAM [XEQ] (F) for closed number)
	Function (monadic)	Integer Part		IP	IP	Integer Part of number (returns Real number) (Shortcut : #TAM [R+] (I) for closed number)
	Flag (system)	Irrational fractions mode preferred	CLEAR	IRFR		SET for irrational fractions mode preferred; CLEAR for irrational fractions mode not preferred (Info : IRFR is SET by IRFRAC and CLEARED by FRACT to preserve state when toggling fraction modes using a b/c and .d)
	Flag	Denominator any	CLEAR	DENANY		SET for any denominator (up to value set interactively from menu DISP (DMX TAM variant) and from menu [INFO.3] fF5 (DMXX)) in fraction display mode; CLEAR for fixed denominator in fraction display mode (Info : DENANY and DENFIX are linked (both cannot be SET at the same time) ; SBI [+ $\times$] is shown if flag SBfrac is SET)
	Flag	Denominator fixed	CLEAR	DENFIX		SET for fixed denominator (set interactively from menu DISP (DMX TAM variant) and from menu [INFO.3] fF5 (DMXX)) in fraction display mode; CLEAR for any denominator in fraction display mode (Info : DENFIX and DENANY are linked (both cannot be SET at the same time) ; SBI [+ $\times$] is shown if flag SBfrac is SET)
	Flag	Fraction mode full cycle	CLEAR	FRCYC		SET for full cycle on key a b/c (g[Σ +]); CLEAR for limited cycle on key a b/c (g[Σ +]) (Info : When FRCYC is SET full cycle is available, when CLEAR, flag PROPFR is excluded (used as is) and OFF state is included in cycle)
	Flag	Fraction show register name	CLEAR	FRCSRN		SET for displaying register name in fractions mode; CLEAR for not displaying register name in fractions mode (Prefix : x, y, z, t < or = or > or $\approx$)
	Flag	Fractions mode	CLEAR	FRACT		SET for fractions mode; CLEAR for fractions mode inactive (Info : FRACT and IRFRAC are linked (both cannot be SET at the same time) ; /n or /max denotes maximum denominator (set interactively from menu DISP (DMX TAM variant) and from menu [INFO.3] fF5 (DMXX)) ; prefixes < and > may be shown as needed for rational fractions ; also see flag IRFRAC; SBI $\approx$ indicates FDIGS < 34 ; SBI is shown if flag SBfrac is SET ; exit fraction modes by .d or CLRMOD (longerpress [EXIT]))
	Flag	Irrational fractions mode	CLEAR	IRFRAC		SET for irrational fractions mode; CLEAR for irrational fractions mode inactive ; IRFRAC gives fractional results only if a simple fraction exists ; on battery power, matrices and vectors are ignored to prevent heavy calculations (Info : IRFRAC and FRACT are linked (both cannot be SET at the same time) ; SBI $\times 1$ or $\cdot 1$ indicates that approximate multiples or fractions can be used, of the irrational numbers $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, $\sqrt{7}$, ϕ , π^{-2} , π^{-1} , $\sqrt{\pi}$, π , π^2 , e^{-2} , e^{-1} , e and e^2 (with tolerance 10^{-24}) ; Prefix $\approx$ + <approximation> ; multiplication symbol SET for displaying multiplication symbol $\times$; CLEAR for displaying multiplication symbol $\cdot$ or SET for displaying multiplication symbol $\times$; CLEAR for displaying multiplication symbol $\cdot$ according to flag MULT $\times$; SBI $\approx$ indicates FDIGS < 34; SBI is shown if flag SBfrac is SET ; exit fraction modes by .d or CLRMOD (longerpress [EXIT]))
	Flag	Proper fractions	SET	PROPFR		SET for proper fractions (mixed fractions); CLEAR for improper fractions
	Command	Get max denominator for fractions	64	DMX#	DMX#	Get max denominator for fractions ; set interactively from menu DISP (DMX TAM variant) and from menu [INFO.3] fF5 (DMXX) (Info : Range is [2, 9999] or 0, indicating DMXmax = 32500 ; SBI is shown if flag SBfrac is SET)
	Command	Get significant digits for fractions	34	FDIGS#	FDIGS#	Get the number of significant digits (1-34) used by the approximations of the fraction modes (Info : Set by FDIGS)
	Command	Set max denominator for fractions	64	DMXX	DMXX ₆₄	Set max denominator for fractions to value in X ; maximum value of 32500 is set by value 0 in X, stored as 0 and displayed as DMXX ₀ ; also set interactively from menu DISP (DMX TAM variant) (Info : Range is [2, 9999] or 0, indicating DMXmax = 32500 ; SBI is shown if flag SBfrac is SET)
Fuel economy conversions	Function (monadic, linked)	km/ℓ to ℓ/100km		km/ℓ→ℓ/100km		Convert kilometre per litre to litre per 100 kilometre (TI : ℓ/100km)
	Function (monadic, linked)	ℓ/100km→km/ℓ		ℓ/100km→km/ℓ		Convert litre per 100 kilometre to kilometre per litre (TI : km/ℓ)
	Function (monadic, linked)	ℓ/100km→mpg _{UK}		ℓ/100km→mpg _{UK}		Convert litre per 100 kilometre to mile per gallon UK (TI : mpg _{UK})
	Function (monadic, linked)	ℓ/100km→mpg _{US}		ℓ/100km→mpg _{US}		Convert litre per 100 kilometre to mile per gallon US (TI : mpg _{US})
	Function (monadic, linked)	mpg _{UK} →ℓ/100km		mpg _{UK} →ℓ/100km		Convert mile per gallon UK to litre per 100 kilometre (TI : ℓ/100km)
	Function (monadic, linked)	mpg _{US} →ℓ/100km		mpg _{US} →ℓ/100km		Convert mile per gallon US to litre per 100 kilometre (TI : ℓ/100km)
Function key	F-key	Function key 1		F1		Function key 1 ; Keyboard [0]
	F-key	Function key 2		F2		Function key 2 ; Keyboard [0]
	F-key	Function key 3		F3		Function key 3 ; Keyboard [0]

Category	Type	FullName	Default	Label	Catalog	Extended description
	F-key	Function key 4		F4		Function key 4 ; Keyboard [0]
	F-key	Function key 5		F5		Function key 5 ; Keyboard [0]
	F-key	Function key 6		F6		Function key 6 ; Keyboard [0]
Humorous conversions	MENU	Humorous conversions		[FFF+]	FFF+	Conversions to and from the furlong-firkin-fortnight (FFF) system (and bananas and beardseconds) -- see https://en.wikipedia.org/wiki/FFF_system (Info : Flag CONV _H may be SET for <to> <f> <to> unit conversion buttons ; CLEAR for <from> >f< <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FFF+_page_1.pdf
Information	MENU	Information		[INFO]	INFO	System information and some information about the value in the X-register ; Keyboard g[.] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_INFO_page_1.pdf
	Command	Data type		TYPE	TYPE	Get datatype of value in X, expressed as a numeric typecode m.nop, where m : 0-9 ; n : 0-5 or 2-15 ; o : 0-4 ; p : 0-2 (TI : <type> ; Info : note that some datatypes have subcodes ; Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command	Last error		LSTERR	LSTERR	Retain error code of last function or command executed ; returns 0 for no error (Ref : https://47calc.com/doc/C47/Reference/C47_Errors_and_messages.pdf ; Info : returns error code even when IGNIER (Ignore one error) is SET)
Input-Output	Setting (pgm, HW)	Set volume	11	VOL	VOL ₁₁	Set volume (0-11) ; hardware only (DMCP setting) (TAM : VOL __ TamNonReg menu ; Info : Error 126 -- Not available on the simulator -- is raised when this setting is performed on the simulator)
	Setting (pgm, HW)	Volume down	11	VOL↓	VOL↓ ₁₁	Set audio volume down (0-11) ; hardware only (DMCP setting) (Info : Error 126 -- Not available on the simulator -- is raised when this setting is performed on the simulator)
	Setting (pgm, HW)	Volume up	11	VOL↑	VOL↑ ₁₁	Set audio volume up (0-11) ; hardware only (DMCP setting) (Info : Error 126 -- Not available on the simulator -- is raised when this setting is performed on the simulator)
	MENU	Audio		[AUDIO]	AUDIO	Audio commands Docs: https://47calc.com/doc/C47/Menus/C47_Menu_AUDIO_page_1.pdf
	MENU	Input/Output		[I/O]	I/O	I/O functions (read/load, write/save, export program, in-export registers, SNAP, audio, print) ; Keyboard g[-] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_I-O_page_1.pdf
	Command (SIM)	Write/save & export all		WRXPall	WRXPall	Write/save & export all programs as binary and rich text files to PROGRAMS/ALLPGMS/ in FAT ; simulator only (Info : TI : Only available on the simulator -- is raised when this command is performed on the hardware)
	Command (nonpgm)	Load full backup		LOAD	LOAD	Load full backup from SAVE file SAVFILES/C47.sav in FAT and auto-clear USER mode ; key assignments will be ignored when loading from a backup file created on a different model (C47 versus R47 and vice versa) (TI : Backup restored + <version info> ; Directory : SAVFILES/)
	Command (nonpgm)	Load programs		LOADP	LOADP	Load programs from SAVE file SAVFILES/C47.sav in FAT (TI : Programs and equations loaded ; appended ; Directory : SAVFILES/)
	Command (nonpgm)	Load registers		LOADR	LOADR	Load registers from SAVE file SAVFILES/C47.sav in FAT (TI : Global and local registers loaded ; (w/ local flags ; Directory : SAVFILES/)
	Command (nonpgm)	Load Sigma registers		LOADΣ	LOADΣ	Load Sigma registers from SAVE file SAVFILES/C47.sav in FAT (TI : Loading stats ... ; Statistical data loaded ; Directory : SAVFILES/)
	Command (nonpgm)	Load state		LOADST	LOADST	Load state from SAVE file (.s47) in FAT and clear USER mode ; if full state, presenting "WARNING: Current calculator state will be lost. Press [ENTER] to confirm." ; key assignments will be ignored when loading from a state file created on a different model (C47 versus R47 and vice versa) (TI : Loading state file ... ; State file loaded ; DMCP File open dialog ("Load Calculator State" -- STATE/))
	Command (nonpgm)	Load system state		LOADSS	LOADSS	Load system state from SAVE file SAVFILES/C47.sav in FAT (TI : System settings loaded ; Directory : SAVFILES/)
	Command (nonpgm)	Load variables		LOADV	LOADV	Load variables from SAVE file SAVFILES/C47.sav in FAT (TI : User variables loaded ; Directory : SAVFILES/)
	Command (nonpgm)	Save full backup		SAVE	SAVE	Save full backup to file SAVFILES/C47.sav in FAT (TI : Saved ; Directory : SAVFILES/)
	Command (nonpgm)	Save full state file		SAVEST	SAVEST	Save backup file (.s47) in FAT for LOADST ; the state file contains the complete state of the calculator: all memory content (registers, variables, programs, equations), all user assignments (key assignments, user menus) and all settings (TI : Saving state file ... ; Saved ; DMCP File save dialog ("Save Calculator State" -- STATE/))
	Command (nonpgm, HW)	Save autoload file		SAVEAUT	SAVEAUT	Save autoload file to file SAVFILES/C47auto.sav in FAT ; hardware only (TI : Saved ; Directory : SAVFILES/ ; Info : Error 126 -- Not available on the simulator -- is raised when this command is performed on the simulator)
	Command (HW)	Buzzer		BUZZ	BUZZ	Play sound with frequency Y in Hz (0 = silent) and duration X in ms (max 2000) ; hardware only (Info : Error 126 -- Not available on the simulator -- is raised when this command is performed on the simulator)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (HW)	Get volume	11	VOL#	VOL#	Get volume (0-11) ; hardware only (DMCP setting) (Info : Set by VOL ; Error 126 -- Not available on the simulator -- is raised when this setting is performed on the simulator)
	Command (HW)	Play (BUZZ) sounds		PLAY	PLAY	Play (BUZZ) sounds using input from nx3 or nx2 matrix variable having rows : [frequency, duration, volume] (Hz, ms, volume element is optional) ; hardware only (TAM : PLAY __ TamLabel menu ; Info : Error 126 -- Not available on the simulator -- is raised when this command is performed on the simulator)
	Command	Beep		BEEP	BEEP	Play beep sound (4 tones)
	Command	Export lettered registers		EXPltr	EXPltr	Export lettered registers A-Z to text data file (.d47) in FAT (Info : DMCP : File save dialog ("Export Register File" -- DATA/) ; TI : Data file saved)
	Command	Export n registers		EXPnrg	EXPnrg	Export n registers starting with R00 to text data file (.d47) in FAT (TAM : EXPnrg ____ TamNonReg menu ; Info : DMCP : File save dialog ("Export Register File" -- DATA/) ; TI : Data file saved)
	Command	Export program		XPORTP	XPORTP	Export program to rich text format file (.rtf) in FAT (TAM : XPORTP __ TamLbOnly menu ; DMCP : File save dialog ("Export Program RTF" -- PROGRAMS/) ; TI : Saved)
	Command	Export register or variable		EXPreg	EXPreg	Export register or variable to text data file (.d47) in FAT (TAM : EXPreg __ Tam menu ; Info : DMCP : File save dialog ("Export Register File" -- DATA/) ; TI : Data file saved)
	Command	Export stack		EXPstk	EXPstk	Export stack (8 or 4 registers, depending on flag SSIZE8) to text data file (.d47) in FAT (Info : DMCP : File save dialog ("Export Register File" -- DATA/) ; TI : Data file saved)
	Command	Export XFN		EXPxfnx	EXPxfnx	Export value of virtual compound data type in XYZ to text data file (.d47) in FAT (Info : DMCP : File save dialog ("Export Register File" -- DATA/) ; TI : Data file saved)
	Command	Import registers		IMPORTr	IMPORTr	Import registers from text data file (.d47), to destination address specified in the file (Info : DMCP : File open dialog ("Import Register File" -- DATA/) ; TI : Data file loaded ; Ref : https://47calc.com/doc/Application%20Notes/AN0025_C47_R47_JM_d47_fileformat.pdf)
	Command	Last statefile loaded		STATE	STATE	Get name of last full statefile or backup loaded from STATE/<state>.s47, SAVFILES/C47.sav or SAVFILES/C47auto.sav (TI : Last full state file loaded: <date> <file>)
	Command	Read/load program		READP	READP	Read/load program from WRITEP backup file (.p47) in FAT (TI : Program file loaded ; DMCP File open dialog ("Load Program" -- PROGRAMS/))
	Command	Screenshot		SNAP	SNAP	Save screenshot to backup file (.bmp) and save contents of stack or alpha buffer as backup file (.REGS.TSV) in FAT ; plays clicking sound (Shortcut : [f/g] + [E] (DMCP-shortcut for screenshot bitmap only) ; Info : screenshot is saved in file SCREENS/<YYYYMMDD-HHMMSS00>.bmp ; SNAP also appends data to file DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; new datafile after timeout of 2 minutes)
	Command	Tone		TONE	TONE	Tone (range 0-9) (TAM : TONE __ TamNonReg menu)
	Command	Write/save program		WRITEP	WRITEP	Write/save program to backup file (.p47) in FAT for READP (TAM : WRITEP __ TamLbOnly menu ; DMCP : File save dialog ("Save Program" -- PROGRAMS/) ; TI : Saved)
Integers	Setting	Display Long integers as Real	OFF	dZ.0	dZ.0 ☐	Switch on displaying Long integers as Real ; not programmable - in PEM, use flag dZ.0 (Info : dZ.0 overrides dLrgLI ; Prefix : Z : ; radix not displayed if DSP = 0 ; larger values are displayed using scientific notation and indicated as Long integer by prefix "Z:" even if dZ.0 is CLEAR ; button shows current radix preference)
	MENU	Integers		[INTS]	INTS	Short integer functions ; Keyboard g[x] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_INTS_page_1.pdf
	Function (triadic)	Double divide		DBL÷	DBL÷	Double word length divide : (Z - least, Y - most significant digits) / (X)
	Function (triadic)	Double remainder		DBLR	DBLR	Double word length remainder
	Function (triadic)	Z~Y modulo X		*MOD	*MOD	Z~Y modulo X
	Function (triadic)	ZxY modulo X		*MOD	*MOD	Z x Y, modulo X
	Function (monadic)	Long integer		LINT	LINT	Convert to Long integer (0) (Info : Max 1000 digits)
	Function (monadic)	Multiply (prime) factors		M.FACT	M.FACT	Multiply (prime) factors from a 2 row matrix in X (row 1 contains the factors, row 2 contains the powers) (Info : E.g. use FACTORS matrix as input in X)
	Function (monadic)	Prime factors		FACTORS	FACTORS	Prime factors of the integer input in X ; factors are calculated in Long integer, stored as Real number (34 digits) and returned as a 2 row matrix (row 1 contains the primes, row 2 contains the powers) ; shows status, number being tested and last factor found ; interrupt by [EXIT] ; registers G, H, K are used to store the last three factors, in Long integer (register K stores the remaining number to be factorised after interrupt) -- see https://en.wikipedia.org/wiki/Pollard's_rho_algorithm ; https://en.wikipedia.org/wiki/Shanks%27s_square_forms_factorization (Info : Pollard's rho algorithm and Shanks's square forms factorization are used ; prime factors will be accurate up to 3 317 044 064 679 887 385 961 980 ; for input values > 34 digits, the long integer factors are saved in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; during execution, status info may be shown: Factors: Small prime trial: p = Factors: Perfect Sq trial: p = Factors: Shanks/Pollard: n = Factors: Shanks/Pollard: n =)
	Function (monadic)	Short integer		SINT	SINT	Convert to Short integer (8.02-8.16)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, cyclic)	To integer		→I	→I	Convert to Long integer (0), Short integer (8.02-8.16) (cyclic, max 1000 digits) (Info : Short integer indicated by subscript ₁₀ ; can show TI: 0vrfl<0: or 0vrfl>64 bits : which can be abbreviated as OF, indicating overflow condition) ; Keyboard f[Σ+]
	Function (dyadic)	Double multiply		DBL×	DBL×	Double word length multiply (result in Y - least and X - most significant digit)
	Function (dyadic)	Greatest common divisor		GCD	GCD	Greatest common divisor of X and Y
	Function (dyadic)	Integer divide		IDIV	IDIV	Integer divide
	Function (dyadic)	Integer divide and remainder		IDIVR	IDIVR	Integer divide (X) and remainder (Y)
	Function (dyadic)	Least common multiple		LCM	LCM	Least common multiple of X and Y
	Function (dyadic)	Remainder		RMD	RMD	Remainder of division of Y by X
	Flag	Display Long integers as Real	CLEAR	dZ.0		SET for displaying Long integers as Real ; CLEAR for not displaying Long integers as Real (Info : dZ.0 overrules dLrgL1 ; Prefix : Z : ; radix not displayed if DSP = 0 ; larger values are displayed using scientific notation and indicated as Long integer by prefix "Z:" even if dZ.0 is CLEAR)
	Command	Get integer sign mode	2	ISM#	ISM#	Get integer sign mode ; set interactively from menu BITSET (1COMPL ; 2COMPL ; SIGNMT ; UNSIGN) and by ISM (Info : 1COMPL: 1 ; 2COMPL: 2 ; SIGNMT: -1 ; UNSIGN: 0)
	Command	Set integer sign mode	2	ISM	ISM ₂	Set integer sign mode to X ; also set interactively from menu BITSET (1COMPL ; 2COMPL ; SIGNMT ; UNSIGN) (Info : 1COMPL: 1 ; 2COMPL: 2 ; SIGNMT: -1 ; UNSIGN: 0)
Integrator	Variable (real)	Accuracy of the integrator	0	ACC	ACC	Reserved Real variable for the accuracy of the integrator (ACC) ; set interactively from menu Toolf (TI : 'ACC' :)
	Variable (real)	Lower limit for integrator	0	↓Lim	↓Lim	Reserved Real variable for the lower limit for integrator (↓Lim) ; displays as ↓L in menu Toolf for larger values ; set interactively from menu Toolf and set by ∫ _Y [∞] (TI : ↓ Lower limit :)
	Variable (real)	Upper limit for integrator	0	↑Lim	↑Lim	Reserved Real variable for the upper limit for integrator (↑Lim) ; displays as ↑L in menu Toolf for larger values ; set interactively from menu Toolf and set by ∫ _Y [∞] (TI : ↑ Upper limit :)
	MENU	Integrator [EQN]		[f f]	∫ f	Integrator (variables) menu for the algebraic expression entered in the equation editor ; enter value and press <var> button(s) to set variables, then press button for variable to be used by the integrator in menu Toolf or this menu ; if only one variable exists it will be selected and starred automatically ; tolerance set by variable ACC (Info : TI : "Select Integrator Variable" is presented if Toolf is activated without a variable having been selected first) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Integral-f_page_1.pdf
	MENU	Integrator tools (RPN)		[Toolf]		Integrator tools and parameter settings (Info : Toolf menu is typically activated by <var> on [f f d (MVAR).1] F2) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ToolInt_ADV_page_1.pdf
	MENU	Integrator tools [EQN]		[Toolf]	Toolf	Integrator tools and parameter settings Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ToolInt_EQN_page_1.pdf
	Flag (system)	Integrator running	CLEAR	INTING		SET for integrator running ; CLEAR for integrator not running
	Command	Integrator (stack limits)		∫ _Y [∞]	∫ _Y [∞]	Solve the integral for the expression entered in the equation editor or the RPN program (with MVAR variables) selected in TAM ; in Program Entry Mode, RPN program must be specified by PGMINT ; integrator limits set from stack (Y, X) ; tolerance set by variable ACC (TAM : ∫ _Y [∞] ___ Tam menu (in PEM, to specify variable to integrate for) ; TI : Accuracy ≈ ; Info : ∫ ≈ displays the accumulated integral sum, the current accuracy achieved, the level counter from 7 down to 0 and an iteration counter ; shown if flag MONIT is SET)
	Command	Integrator (variable limits)		∫	∫ f d	Solve the integral for the expression entered in the equation editor or the RPN program (with MVAR variables) selected in TAM ; in Program Entry Mode, RPN program must be specified by PGMINT ; integrator limits set by variables (↓Lim, ↑Lim) ; tolerance set by variable ACC (TAM : ∫ ___ Tam menu (in PEM, to specify variable to integrate for) ; TI : Accuracy ≈ ; Info : ∫ ≈ displays the accumulated integral sum, the current accuracy achieved, the level counter from 7 down to 0 and an iteration counter ; shown if flag MONIT is SET)
Keyboard	MENU	Access blue functions		[BLUE47]	BLUE47	Access all functions of the DM42-layout g-shift keyboard (blue) and a few of the f-shift keyboard (Info : Assigned to calculator function SETUP (f[0]) in layout DM42) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_BLUE47_page_1.pdf
	g-shift	Shift g		g	g	Shift g (blue) (Shortcut : double [f/g] ; longpress [f/g] ; Info : SBI position depends on flag SBshfR)
	f/g-shift	Shift f/g		f/g	f/g	Single press : shift f (yellow) ; double press : shift g (blue) (Info : SBI f ; g is shown if flag SBshfR is SET ; note that USER mode assignment to longerpress f/g is possible, if either HOME [®] or MyM [®] is SET, see Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard [f/g]
	f-shift	Shift f		f	f	Shift f (yellow) (Info : SBI position depends on flag SBshfR)
Lettered flags	Flag (lettered)	Flag A		A		Flag A (104) (Info : Reached from FF [FLG.1] fF3 or FF {A})
	Flag (lettered)	Flag B		B		Flag B (105) (Info : Reached from FF [FLG.1] fF4 or FF {B})
	Flag (lettered)	Flag C		C		Flag C (106) (Info : Reached from FF [FLG.1] fF5 or FF {C})
	Flag (lettered)	Flag D		D		Flag D (107) (Info : Reached from FF [FLG.1] fF6 or FF {D})

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag (lettered)	Flag E		E		Flag E (217) (Info : Reached from FF [FLG.2]F1 or FF {E})
	Flag (lettered)	Flag F		F		Flag F (218) (Info : Reached from FF [FLG.2]F2)
	Flag (lettered)	Flag G		G		Flag G (219) (Info : Reached from FF [FLG.2]F3 or FF {G})
	Flag (lettered)	Flag H		H		Flag H (220) (Info : Reached from FF [FLG.2]F4 or FF {H})
	Flag (lettered)	Flag I		I		Flag I (109) (Info : Reached from FF [FLG.1]gF1 or FF {I})
	Flag (lettered)	Flag J		J		Flag J (110) (Info : Reached from FF [FLG.1]gF2 or FF {J})
	Flag (lettered)	Flag K		K		Flag K (111) (Info : Reached from FF [FLG.1]fF1 or FF {K})
	Flag (lettered)	Flag L		L		Flag L (108) (Info : Reached from FF [FLG.1]fF2 or FF {L})
	Flag (lettered)	Flag M		M		Flag M (211) (Info : Reached from FF [FLG.1]F1 or FF {M})
	Flag (lettered)	Flag N		N		Flag N (212) (Info : Reached from FF [FLG.1]F2 or FF {N})
	Flag (lettered)	Flag O		O		Flag O (221) (Info : Reached from FF [FLG.2]F5 or FF {O})
	Flag (lettered)	Flag P		P		Flag P (213) (Info : Reached from FF [FLG.1]F3)
	Flag (lettered)	Flag Q		Q		Flag Q (214) (Info : Reached from FF [FLG.1]F4)
	Flag (lettered)	Flag R		R		Flag R (215) (Info : Reached from FF [FLG.1]F5)
	Flag (lettered)	Flag S		S		Flag S (216) (Info : Reached from FF [FLG.1]F6)
	Flag (lettered)	Flag T		T		Flag T (103) (Info : Reached from FF [FLG.1]gF6)
	Flag (lettered)	Flag U		U		Flag U (222) (Info : Reached from FF [FLG.2]F6)
	Flag (lettered)	Flag V		V		Flag V (223) (Info : Reached from FF [FLG.2]fF1)
	Flag (lettered)	Flag W		W		Flag W (224) (Info : Reached from FF [FLG.2]fF2)
	Flag (lettered)	Flag X		X		Flag X (100) (Info : Reached from FF [FLG.1]gF3)
	Flag (lettered)	Flag Y		Y		Flag Y (101) (Info : Reached from FF [FLG.1]gF4)
	Flag (lettered)	Flag Z		Z		Flag Z (102) (Info : Reached from FF [FLG.1]gF5)
Lettered registers	Register (lettered)	Register A		A		Register A (104) : Stack level 5 (Info : Reached using stack or from STO/RCL [REG.1]fF3 or STO/RCL {A})
	Register (lettered)	Register B		B		Register B (105) : Stack level 6 (Info : Reached using stack or from STO/RCL [REG.1]fF4 or STO/RCL {B})
	Register (lettered)	Register C		C		Register C (106) : Stack level 7 (Info : Reached using stack or from STO/RCL [REG.1]fF5 or STO/RCL {C})
	Register (lettered)	Register D		D		Register D (107) : Stack level 8 (Info : Reached using stack or from STO/RCL [REG.1]fF6 or STO/RCL {D})
	Register (lettered)	Register E		E		Register E (217) : General purpose register E (Info : Reached from STO/RCL [REG.2]F1 or STO/RCL {E})
	Register (lettered)	Register F		F		Register F (218) : General purpose register F (Info : Reached from STO/RCL [REG.2]F2)
	Register (lettered)	Register G		G		Register G (219) : General purpose register G ; also used to store third to last long integer prime factor for FACTORS (Info : Reached from STO/RCL [REG.2]F3 or STO/RCL {G})
	Register (lettered)	Register H		H		Register H (220) : General purpose register H ; also used to store second to last long integer prime factor for FACTORS (Info : Reached from STO/RCL [REG.2]F4 or STO/RCL {H})
	Register (lettered)	Register I		I		Register I (109) : General purpose register I ; also used to store current row of indexed matrix (Info : Reached from STO/RCL [REG.1]gF1 or STO/RCL {I})
	Register (lettered)	Register J		J		Register J (110) : General purpose register J ; also used to store current column of indexed matrix (Info : Reached from STO/RCL [REG.1]gF2 or STO/RCL {J})
	Register (lettered)	Register K		K		Register K (111) : General purpose register K ; also used in VarMNU to store variable name when MVAR-defined button (<var...>) is pressed without NIM input ; also used to store last long integer prime factor for FACTORS or the remaining number to be factorised ; also used as the default alpha register for commands compatible with the HP-42S/DM42(n) calculator (Info : Reached from STO/RCL [REG.1]fF1 or STO/RCL {K})
	Register (lettered)	Register L		L		Register L (108) : Last X (Info : Reached from STO/RCL [REG.1]fF2 or STO/RCL {L})
	Register (lettered)	Register M		M		Register M (211) : General purpose register M ; also used to store distribution parameter : mean, median, location, degree of freedom 1, centre, lower limit (Info : Reached from STO/RCL [REG.1]F1 or STO/RCL {M})
	Register (lettered)	Register N		N		Register N (212) : General purpose register N ; also used to store distribution parameter : degree of freedom 2, quantity, upper limit (Info : Reached from STO/RCL [REG.1]F2 or STO/RCL {N})
	Register (lettered)	Register O		O		Register O (221) : General purpose register O (Info : Reached from STO/RCL [REG.2]F5 or STO/RCL {O})
	Register (lettered)	Register P		P		Register P (213) : General purpose register P ; also used to store distribution parameter : probability (Info : Reached from STO/RCL [REG.1]F3)
	Register (lettered)	Register Q		Q		Register Q (214) : General purpose register Q ; also used to store distribution parameter : shape, 3rd integer (Info : Reached from STO/RCL [REG.1]F4)
	Register (lettered)	Register R		R		Register R (215) : General purpose register R ; also used to store distribution parameter : rate (Info : Reached from STO/RCL [REG.1]F5)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Register (lettered)	Register S		S		Register S (216) : General purpose register S ; also used to store distribution parameter : scale, spread (Info : Reached from ST0/RCL [REG.1]F6)
	Register (lettered)	Register T		T		Register T (103) : Stack level 4 (Info : Reached using stack or from ST0/RCL [Tam.1]F6)
	Register (lettered)	Register U		U		Register U (222) : General purpose register U (Info : Reached from ST0/RCL [REG.2]F6)
	Register (lettered)	Register V		V		Register V (223) : General purpose register V (Info : Reached from ST0/RCL [REG.2]fF1)
	Register (lettered)	Register W		W		Register W (224) : General purpose register W (Info : Reached from ST0/RCL [REG.2]fF2)
	Register (lettered)	Register X		X		Register X (100) : Stack level 1 (Info : Reached using stack or from ST0/RCL [Tam.1]F3)
	Register (lettered)	Register Y		Y		Register Y (101) : Stack level 2 (Info : Reached using stack or from ST0/RCL [Tam.1]F4)
	Register (lettered)	Register Z		Z		Register Z (102) : Stack level 3 (Info : Reached using stack or from ST0/RCL [Tam.1]F5)
Linear Force Density conversions	Function (monadic, linked)	kgf/m to N/m		kgf/m→N/m		Convert kilogram force per metre to Newton per metre (TI : N/m)
	Function (monadic, linked)	lbf/ft to N/m		lbf/ft→N/m		Convert pound force per foot to Newton per metre (TI : N/m)
	Function (monadic, linked)	N/m to kgf/m		N/m→kgf/m		Convert Newton per metre to kilogram force per metre (TI : kgf/m)
	Function (monadic, linked)	N/m to lbf/ft		N/m→lbf/ft		Convert Newton per metre to pound force per foot (TI : lbf/ft)
Mass conversions	MENU	Mass conversions		[Mass:]	Mass:	Convert between units of mass (Info : Flag CONV _{HP} may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> → ← <from> unit conversion buttons (default SET) ; Ref : https://47calc.com/doc/Application%20Notes/AN0027_C47_R47_DA_Structural_conversions.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Mass_page_1.pdf
	Function (monadic, linked)	blob to kg		blob→kg		Convert blob to kilogram ; this is the same conversion as lbf-s ² /in→kg on [Mass.2]gF3 (TI : kg)
	Function (monadic, linked)	carat to g		carat→g		Convert carat to gram (TI : g)
	Function (monadic, linked)	cwt to kg		cwt→kg		Convert hundredweight to kilogram (TI : kg)
	Function (monadic, linked)	fir to kg		fir→kg		Convert firkin to kilogram (TI : kg)
	Function (monadic, linked)	fir to lbs		fir→lbs		Convert firkin to pound (TI : lbs)
	Function (monadic, linked)	g to carat		g→carat		Convert gram to carat (TI : carat)
	Function (monadic, linked)	g to oz		g→oz		Convert gram to ounce (TI : oz)
	Function (monadic, linked)	g to tr oz		g→tr oz		Convert gram to troy ounce (TI : tr oz)
	Function (monadic, linked)	jīn to kg		jīn→kg		Convert jīn to kilogram (TI : kg)
	Function (monadic, linked)	kg to blob		kg→blob		Convert kilogram to blob ; this is the same conversion as kg→lbf-s ² /in on [Mass.2]gF4 (TI : blob)
	Function (monadic, linked)	kg to cwt		kg→cwt		Convert kilogram to hundredweight (TI : cwt)
	Function (monadic, linked)	kg to fir		kg→fir		Convert kilogram to firkin (TI : fir)
	Function (monadic, linked)	kg to jīn		kg→jīn		Convert kilogram to jīn (TI : jīn)
	Function (monadic, linked)	kg to lbf-s ² /ft		kg→lbf-s ² /ft		Convert kilogram to lbf-s ² /ft (slug) ; this is the same conversion as kg→slug on [Mass.1]gF2 (TI : lbf-s ² /ft ; Info : single = bf-s ² /ft)
	Function (monadic, linked)	kg to lbf-s ² /in		kg→lbf-s ² /in		Convert kilogram to lbf-s ² /in (slinch or blob) ; this is the same conversion as kg→slinch on [Mass.1]gF4 (TI : lbf-s ² /in ; Info : single = bf-s ² /in)
	Function (monadic, linked)	kg to lbs		kg→lbs		Convert kilogram to pound (TI : lbs)
	Function (monadic, linked)	kg to liǎng		kg→liǎng		Convert kilogram to liǎng (TI : liǎng)
	Function (monadic, linked)	kg to long ton		kg→long ton		Convert kilogram to long ton (TI : long ton ; Info : single = ng ton)
	Function (monadic, linked)	kg to short cwt		kg→short cwt		Convert kilogram to short hundredweight (TI : short cwt ; Info : single = rt cwt)
	Function (monadic, linked)	kg to short ton		kg→short ton		Convert kilogram to short ton (TI : short ton ; Info : single = rt ton)
	Function (monadic, linked)	kg to slinch		kg→slinch		Convert kilogram to slinch ; this is the same conversion as kg→lbf-s ² /in on [Mass.2]gF4 (TI : slinch)
	Function (monadic, linked)	kg to slug		kg→slug		Convert kilogram to slug ; this is the same conversion as kg→lbf-s ² /ft on [Mass.2]gF2 (TI : slug)
	Function (monadic, linked)	kg to stone		kg→stone		Convert kilogram to stone (TI : stone)
	Function (monadic, linked)	kg to tonne		kg→tonne		Convert kilogram to metric tonne (TI : tonne)
	Function (monadic, linked)	lbf-s ² /ft to kg		lbf-s ² /ft→kg		Convert lbf-s ² /ft (slug) to kilogram ; this is the same conversion as slug→kg on [Mass.1]gF1 (TI : kg)
	Function (monadic, linked)	lbf-s ² /in to kg		lbf-s ² /in→kg		Convert lbf-s ² /in (slinch or blob) to kilogram ; this is the same conversion as slinch→kg on [Mass.1]gF3 (TI : kg)
	Function (monadic, linked)	lbs to fir		lbs→fir		Convert pound to firkin (TI : fir)
	Function (monadic, linked)	lbs to kg		lbs→kg		Convert pound to kilogram (TI : kg)
	Function (monadic, linked)	liǎng to kg		liǎng→kg		Convert liǎng to kilogram (TI : kg)
	Function (monadic, linked)	long ton to kg		long ton→kg		Convert long ton to kilogram (TI : kg)
	Function (monadic, linked)	oz to g		oz→g		Convert ounce to gram (TI : g)
	Function (monadic, linked)	short cwt to kg		short cwt→kg		Convert short hundredweight to kilogram (TI : kg)
	Function (monadic, linked)	short ton to kg		short ton→kg		Convert short ton to kilogram (TI : kg)
	Function (monadic, linked)	slinch to kg		slinch→kg		Convert slinch to kilogram ; this is the same conversion as lbf-s ² /in→kg on [Mass.2]gF3 (TI : kg)
	Function (monadic, linked)	slug to kg		slug→kg		Convert slug to kilogram ; this is the same conversion as lbf-s ² /ft→kg on [Mass.2]gF1 (TI : kg)
	Function (monadic, linked)	stone to kg		stone→kg		Convert stone to kilogram (TI : kg)
	Function (monadic, linked)	tonne to kg		tonne→kg		Convert metric tonne to kilogram (TI : kg)
	Function (monadic, linked)	tr oz to g		tr oz→g		Convert troy ounce to gram (TI : g)

Category	Type	FullName	Default	Label	Catalog	Extended description
Mathematics	Variable (real)	Delta value for derivatives		δ_d	δ_d	Delta value for derivatives, used as the step size (Info : After a value has been entered, it is show as a subscript ; if no value is entered, or a variable 'δ _d ' has already been defined by the user, an automatic step size is determined by iteration in 16 steps, starting with 1/10 ; the step size will always be displayed as "δ")
	Setting (pgm)	Set significant digits	34	SDIGS	SDIGS ₃₄	Set the number of significant digits (1-34) for rounding after each operation ; sets tolerance of solvers, plotting and CONVg? ; value of 0 sets maximum precision (34) (TAM : SDIGS ___ TamNonReg menu ; Info : when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS)
	Setting (pgm, old)	Set significant digits	34	>SDIGS<		Old version of the setting SDIGS that used stack input (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use SDIGS on [MODE.1] F4 instead)
	Setting	Special results	ON	SPCRES	SPCRES $\blacksquare$	Switch on special results ; not programmable - in PEM, use flag SPCRES (Info : Special results are: infinity ($\pm\infty$) and not-a-number (NaN), an error will not occur for such values when SPCRES is SET ; also, ± 0 . may be used)
	MENU	Elliptical functions		[Ellipt]	ELLIPT	Elliptical functions (Ref : https://47calc.com/doc/Application%20Notes/AN0023_C47_R47_WGR_JM_DE_Elliptic_Incl_Anxi1.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Ellipt_page_1.pdf
	MENU	Exponential		[EXP]	EXP	Exponential functions ; Keyboard g[E] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_EXP_page_1.pdf
	MENU	Extended functions		[X.FN]	X.FN	Extended functions (Bessel, Bernoulli, Gamma, Elliptical, Orthogonal, etcetera) (Ref : https://47calc.com/doc/C47/Reference/C47_Prime_numbers.pdf) ; Keyboard g[9] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_X-FN_page_1.pdf
	MENU	Number theory functions		[NumTh]	NumTh	Number theory functions -- see https://en.wikipedia.org/wiki/Divisor_function Docs: https://47calc.com/doc/C47/Menus/C47_Menu_NumTh_page_1.pdf
	MENU	Orthogonal polynomials		[Orthog]	ORTHOG	Orthogonal polynomials Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Orthog_page_1.pdf
	Function (triadic)	Laguerre generalised polynomials		$L_m^{(\alpha)}$	$L_m^{(\alpha)}$	Laguerre generalised polynomials (Parameter : x = X ; m = Y ; α = Z)
	Function (triadic)	Linear interpolation		LINPOL	LINPOL	Linear interpolation ; also works for Complex numbers and Dates (Info : <from> ENTER <towards> ENTER <at> (fraction) LINPOL)
	Function (triadic)	Regularised (incomplete) Beta function		I_{xyz}	I_{xyz}	Regularised (incomplete) Beta function -- see https://en.wikipedia.org/wiki/Beta_function
	Function (monadic)	$\sqrt{1+x^2}$		$\sqrt{1+x^2}$	$\sqrt{1+x^2}$	Calculate $\sqrt{1+x^2}$
	Function (monadic)	10 to the power x		10^x	10^x	Raise 10 to the power in the X-register ; Keyboard f[LOG]
	Function (monadic)	2 to the power x		2^x	2^x	Raise 2 to the power in the X-register
	Function (monadic)	Angle to modulus		$\theta \rightarrow m$	$\theta \rightarrow m$	Ellipse eccentricity angle to modulus : m = $\sin^2(\theta)$ (TI : modulus m=k ² :)
	Function (monadic)	Bernoulli number (new definition)		B_n	B_n	Bernoulli number (new definition)
	Function (monadic)	Bernoulli number (old definition)		B_n	B_n	Bernoulli number (old definition)
	Function (monadic)	Bessel function of the 1st kind		$J_y(x)$	$J_y(x)$	Bessel function of the 1st kind and order y -- see http://en.wikipedia.org/wiki/Bessel_function
	Function (monadic)	Binary logarithm		LB	LB	Binary logarithm (base 2)
	Function (monadic)	Common logarithm		LOG	LOG	Common logarithm (base 10) ; Keyboard [LOG]
	Function (monadic)	Complementary error function		erfc	erfc	Complementary error function
	Function (monadic)	Complete elliptic integral (1st)		K(m)	K(m)	Complete elliptic integral of the 1st kind -- see https://mathworld.wolfram.com/CompleteEllipticIntegraloftheFirstKind.html (Parameter : m = X)
	Function (monadic)	Complete elliptic integral (2nd)		E(m)	E(m)	Complete elliptic integral of the 2nd kind -- see https://mathworld.wolfram.com/CompleteEllipticIntegraloftheSecondKind.html (Parameter : m = X)
	Function (monadic)	Cosine of πx		$(-1)^x$	$(-1)^x$	Cosine of πx ; for complex parameter : $(-1)^x = e^{i\pi x} = \cos(\pi x) + i\sin(\pi x)$ (Info : Esp. for non-integer x)
	Function (monadic)	Cube		x^3	x^3	Raise to power of 3
	Function (monadic)	Cube root		$\sqrt[3]{x}$	$\sqrt[3]{x}$	Cube root of X
	Function (monadic)	Divisor count function (k = 0)		σ_0	σ_0	Divisor count function (k = 0) using either a FACTORS matrix or Long integer as input in X
	Function (monadic)	e to the power x		e^x	e^x	Raise e to the power in the X-register ; Keyboard f[LN]
	Function (monadic)	Eccentricity to modulus		$k \rightarrow m$	$k \rightarrow m$	Ellipse eccentricity to modulus : m = k ² (TI : modulus m=k ² :)
	Function (monadic)	Error function		erf	erf	Error function
	Function (monadic)	Euler's formula		e^{ix}	e^{ix}	Rotate complex unit vector by X radians : $e^{ix} = \cos(x) + i\sin(x)$ (Info : i or j displayed in stack and on soft buttons)
	Function (monadic)	Euler's totient function		ϕ_e	ϕ_e	Euler's totient function using either a FACTORS matrix or Long integer as input in X -- see https://en.wikipedia.org/wiki/Euler's_totient_function
	Function (monadic)	$e^x - 1$		$e^x - 1$	$e^x - 1$	More accurate calculation of $e^x - 1$ for x $\approx$ 0
	Function (monadic)	Factorial x! or $\Gamma(x+1)$		x!	x!	For Long integers : x! ; for Reals : $\Gamma(x+1)$ (Info : Max integer: 450 (up to 1000 digits accurate) ; max real : 2123.549 956 662 463 236 31 ; integers > max are converted to reals)
	Function (monadic)	Fibonacci number		FIB	FIB	Fibonacci number n, where n = X -- see https://en.wikipedia.org/wiki/Fibonacci_sequence
	Function (monadic)	Gamma function		$\Gamma(x)$	$\Gamma(x)$	Gamma function
	Function (monadic)	Gudermannian function		gd	gd	Gudermannian function -- see http://en.wikipedia.org/wiki/Gudermannian_function

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic)	Hyperbolic cosine		cosh	cosh	Hyperbolic cosine
	Function (monadic)	Hyperbolic sine		sinh	sinh	Hyperbolic sine
	Function (monadic)	Hyperbolic tangent		tanh	tanh	Hyperbolic tangent
	Function (monadic)	Inverse Gudermannian function		gd ¹	gd ¹	Inverse Gudermannian function -- see http://en.wikipedia.org/wiki/Gudermannian_function
	Function (monadic)	Inverse hyperbolic cosine		arcosh	arcosh	Inverse hyperbolic cosine
	Function (monadic)	Inverse hyperbolic sine		arsinh	arsinh	Inverse hyperbolic sine
	Function (monadic)	Inverse hyperbolic tangent		artanh	artanh	Inverse hyperbolic tangent
	Function (monadic)	Inverse of Lambert's W function (principal)		W ⁻¹	W ⁻¹	Inverse of Lambert's W function (principal branch, ≥ -1)
	Function (monadic)	Lambert's W function (negative)		W _m	W _m	Lambert's W function (negative branch) ; m = minus -- see http://en.wikipedia.org/wiki/Lambert_W_function
	Function (monadic)	Lambert's W function (principal)		W _p	W _p	Lambert's W function (principal branch) -- see http://mathworld.wolfram.com/LambertW-Function.html
	Function (monadic)	LN(1+x)		LN(1+x)	LN(1+x)	More accurate calculation of LN(1+x) for $x \approx 0$
	Function (monadic)	Modulus to angle		m $\rightarrow$ θ	m $\rightarrow$ θ	Ellipse modulus to eccentricity angle : $\theta = \text{ARCSIN}(\sqrt{m})$ (TI : eccentricity angle θ :)
	Function (monadic)	Modulus to eccentricity		m $\leftarrow$ k	m $\leftarrow$ k	Ellipse modulus to eccentricity : $k = \sqrt{m}$ (TI : eccentricity $e=k=\sqrt{m}$:)
	Function (monadic)	Natural logarithm of Gamma function		LN Γ	LN Γ	Natural logarithm of Gamma function
	Function (monadic)	Natural logarithm		LN	LN	Natural logarithm (base e) ; Keyboard [LN]
	Function (monadic)	Next prime		NEXTP	NEXTP	Next prime of the integer X (Info : Prime numbers are calculated only if $X < 10^{308}$, or NEXTP raises Error 1 -- An argument exceeds the function domain (may depend on flag CPXRES and SPCRES))
	Function (monadic)	Polynomial 4th degree		Poly4		Create polynomial : $a4 \times x^4 + a3 \times x^3 + a2 \times x^2 + a1 \times x + a0$
	Function (monadic)	Proper sigma function		σ_1^+	σ_1^+	Proper sigma function using either a FACTORS matrix or Long integer as input in X
	Function (monadic)	Reciprocal		1/x	1/x	Reciprocal (1/X) (Info : When X is a matrix 1/x inverts it ([M] ⁻¹)) ; Keyboard [1/x]
	Function (monadic)	Riemann's Zeta for real arguments		$\zeta(x)$	$\zeta(x)$	Riemann's Zeta for real arguments -- see http://mathworld.wolfram.com/RiemannZetaFunction.html
	Function (monadic)	Sigma function (k = 1)		σ_1	σ_1	Sigma function (k = 1) using either a FACTORS matrix or Long integer as input in X
	Function (monadic)	Sinc		sinc	sinc	Calculate (Sine of X) / X (Info : Integers and untagged reals are assumed to be radians)
	Function (monadic)	Sinc pi		sinc π	sinc π	Calculate (Sine of $\pi \times X$) / ($\pi \times X$) (Info : Integers and untagged reals are assumed to be radians)
	Function (monadic)	Square		x ²	x ²	Square of X ; Keyboard f[$\sqrt{x}$]
	Function (monadic)	Square root		$\sqrt{x}$	$\sqrt{x}$	Square root of X (Info : When X is a matrix $\sqrt{x}$ calculates its square root (MSQRT)) ; Keyboard [$\sqrt{x}$]
	Function (monadic)	Unit in the last place	10 ⁻⁶¹⁷⁶	ULP	ULP	Get minimum difference to next or previous machine representable real, as power of ten (Info : ULP is for Real numbers in X ; result is displayed as ≈ 0 . because of default HIDE and RNG settings, use SHOW to view result)
	Function (monadic, legacy)	Square root		>SQRT<		Legacy version of function $\sqrt{x}$: Square root of X (Info : Function >SQRT< is superseded by $\sqrt{x}$ on [$\sqrt{x}$])
	Function (dyadic)	Arithmetic geometric mean		AGM	AGM	Arithmetic geometric mean of X and Y
	Function (dyadic)	Chebyshev polynomials (1st)		T _n	T _n	Chebyshev polynomials of the 1st kind (Parameter : x = X ; n = Y)
	Function (dyadic)	Chebyshev polynomials (2nd)		U _n	U _n	Chebyshev polynomials of the 2nd kind (Parameter : x = X ; n = Y)
	Function (dyadic)	Complete elliptic integral (2rd)		$\Pi(n,m)$	$\Pi(n,m)$	Complete elliptic integral of the 3rd kind (Parameter : n = X ; m = Y)
	Function (dyadic)	Dimensions to modulus		a,b $\rightarrow$ m	a,b $\rightarrow$ m	Ellipse dimensions to modulus : $m = 1-(a/b)^2$, where b is the larger axis (TI : modulus $m=k^2$:)
	Function (dyadic)	Divide		$\div$	$\div$	Divide Y by X ; Keyboard [$\div$]
	Function (dyadic)	Elliptic amplitude		am(u,m)	am(u,m)	Elliptic amplitude (Parameter : u = X ; m = Y ; Info : u is a Real number (1.0), not an Angle (1.1-1.5))
	Function (dyadic)	Elliptic cosine		cn(u,m)	cn(u,m)	Elliptic cosine (Parameter : u = X ; m = Y ; Info :)
	Function (dyadic)	Elliptic delta amplitude		dn(u,m)	dn(u,m)	Elliptic delta amplitude (Parameter : u = X ; m = Y ; Info : u is a Real number (1.0), not an Angle (1.1-1.5))
	Function (dyadic)	Elliptic sine		sn(u,m)	sn(u,m)	Elliptic sine (Parameter : u = X ; m = Y ; Info : u is Real number (1.0), not an Angle (1.1-1.5))
	Function (dyadic)	Euler's Beta function		$\beta(x,y)$	$\beta(x,y)$	Euler's Beta function
	Function (dyadic)	Generalised proper sigma function		σ_k^+	σ_k^+	Generalised proper sigma function using either a FACTORS matrix or Long integer as input in X
	Function (dyadic)	Generalised sigma function		σ_k	σ_k	Generalised sigma function using either a FACTORS matrix or Long integer as input in X
	Function (dyadic)	Hermite polynomials (physics)		H _{nP}	H _{nP}	Hermite polynomials (physics) (Parameter : x = X ; n = Y)
	Function (dyadic)	Hermite polynomials (probability)		H _n	H _n	Hermite polynomials (probability) (Parameter : x = X ; n = Y)
	Function (dyadic)	Incomplete elliptic integral (1st)		F(ϕ ,m)	F(ϕ ,m)	Incomplete elliptic integral of the 1st kind (Parameter : ϕ = X ; m = Y ; Info : ϕ is amplitude angle (any angle mode))
	Function (dyadic)	Incomplete elliptic integral (2nd)		E(ϕ ,m)	E(ϕ ,m)	Incomplete elliptic integral of the 2nd kind (Parameter : ϕ = X ; m = Y ; Info : ϕ is amplitude angle (any angle mode))
	Function (dyadic)	Jacobi's Zeta		Z(ϕ ,m)	Z(ϕ ,m)	Jacobi's Zeta (Parameter : ϕ = X ; m = Y ; Info : ϕ is amplitude angle (any angle mode))
	Function (dyadic)	Laguerre polynomials		L _m	L _m	Laguerre polynomials (Parameter : x = X ; m = Y)
	Function (dyadic)	Legendre polynomials		P _n	P _n	Legendre polynomials (Parameter : x = X ; n = Y)
	Function (dyadic)			L0Gxy	L0Gxy	Logarithm of Y for base X
	Function (dyadic)	Lower incomplete Gamma function		γ_{xy}	γ_{xy}	Lower incomplete Gamma function -- see https://en.wikipedia.org/wiki/Incomplete_gamma_function

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (dyadic)	Lower regularised incomplete Gamma function		Γ_p	Γ_p	Lower regularised incomplete Gamma function (P) -- see https://mathworld.wolfram.com/RegularizedGammaFunction.html
	Function (dyadic)	Maximum		MAX	MAX	Maximum of X and Y
	Function (dyadic)	Minimum		MIN	MIN	Minimum of X and Y
	Function (dyadic)	Modulo		MOD	MOD	Y modulo X
	Function (dyadic)	Multiply		$\times$	$\times$	Multiply Y by X ; Keyboard [x]
	Function (dyadic)	Natural logarithm of Euler's Beta function		LN β	LN β	Natural logarithm of Euler's Beta function
	Function (dyadic)	Neighbour		NEIGHB	NEIGHB	Get neighbour value of X compared to Y (next integer or next machine representable Real number)
	Function (dyadic)	Subtract		-	-	Subtract X from Y ; Keyboard [-]
	Function (dyadic)	Upper incomplete Gamma function		Γ_{xy}	Γ_{xy}	Upper incomplete Gamma function -- see https://en.wikipedia.org/wiki/Incomplete_gamma_function
	Function (dyadic)	Upper regularised incomplete Gamma function		Γ_q	Γ_q	Upper regularised incomplete Gamma function (Q) -- see https://mathworld.wolfram.com/RegularizedGammaFunction.html
	Function (dyadic)	xth root		$\sqrt[x]{y}$	$\sqrt[x]{y}$	Xth root of Y ; Keyboard g[R4]
	Function (dyadic)	y to the power x		y^x	y^x	Raise value in the Y-register to the power in the X-register ; Keyboard f[1/x]
	Flag	Special results	SET	SPCRES		SET for special results; CLEAR for special results inactive (Info : Special results are: infinity ($\pm\infty$) and not-a-number (NaN), an error will not occur for such values when SPCRES is SET ; also, ± 0 . may be used)
	Command	Bessel function of the 2nd kind		$Y_y(x)$	$Y_y(x)$	Bessel function of the 2nd kind and order y -- see http://en.wikipedia.org/wiki/Bessel_function
	Command	Get significant digits	34	SDIGS#	SDIGS#	Get the number of significant digits (1-34) for rounding after each operation (Info : Set by SDIGS)
Mathematics and characters	Symbol	Bar				Open or close absolute value (Shortcut : alpha g[ST0] ; Info : Fontcode : 007C ₁₆)
	Symbol	Common logarithm		LOG		Common logarithm (base 10) (Shortcut : alpha g[LOG] ("LOG"))
	Symbol	Cube root		$\sqrt[3]{}$		Cube root (Fontcode : 221B ₁₆)
	Symbol	e to the power x		e^x		e to the power x
	Symbol	Equal		=		Equal (Info : Fontcode : 003D ₁₆)
	Symbol	Exclamation mark		!		In edit equation, used as Factorial x! or $\Gamma(x+1)$ (Shortcut : alpha f[R/S] ; Info : Fontcode : 0021 ₁₆)
	Symbol	Integral		$\int$		Integral (Fontcode : 222B ₁₆)
	Symbol	Left parenthesis		(		Open (left) parenthesis (Fontcode : 0028 ₁₆)
	Symbol	Natural logarithm		LN		Natural logarithm (base e) (Shortcut : alpha g[LN] ("LN"))
	Symbol	pi		π		Greek character π (Shortcut : alpha g[R+] ; Info : Fontcode : 03C0 ₁₆ ; press [▲] for Π at [A...Q.1] gF5)
	Symbol	Raise to power		$\wedge$		Raise to power (Shortcut : alpha g[1/x] ; Info : Fontcode : 005E ₁₆)
	Symbol	Right parenthesis		)		Close (right) parenthesis (Fontcode : 0029 ₁₆)
	Symbol	Square root		$\sqrt{}$		Square root (Shortcut : alpha g[√x] ; Info : Fontcode : 221A ₁₆)
	Character	Square root		$\sqrt{}$		Square root (Fontcode : 221A ₁₆)
Mathematics and text functions	Function (dyadic)	Add		+	+	Add X to Y ; concatenate X and Y, including numbers, dates and times + strings and vice versa ; Keyboard [+]
Matrix operations	Variable (matrix)	Matrix A		Mat A	Mat_A	Reserved Matrix variable for the Matrix A (Mat A) ; enter equation coefficient matrix using menu M_SIMQ ; created after RESET
	Variable (matrix)	Matrix B		Mat B	Mat_B	Reserved Matrix variable for the Matrix B (Mat B) ; enter equation result vector using menu M_SIMQ ; created after RESET
	Variable (matrix)	Matrix X		Mat X	Mat_X	Reserved Matrix variable for the Matrix X (Mat X) ; calculate equation solution vector using menu M_SIMQ ; created after RESET
	Setting	Grow (matrix edit)	OFF	GROW	M.GROW o	Switch on grow (matrix edit) ; not programmable - in PEM, use flag GROW (Info : SBI is shown if flag SBmx is SET)
	Setting	Grow (matrix edit)	ON	WRAP	M.WRAP @	Switch on wrap (matrix edit) ; not programmable - in PEM, use flag GROW (Info : SBI is shown if flag SBmx is SET)
	MENU (item)	Matrix simultaneous equations		[SIM EQ]	SIM_EQ	Matrix simultaneous equations : Mat A · Mat X = Mat B, where Mat A is the equation coefficient matrix and Mat B is the equation result vector ; Mat X is the equation solution vector (TAM : SIM_EQ ... TamNonReg menu to enter number of equations ; Info : CAT.MENUS M_SIMQ) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_SIM_EQ_page_1.pdf
	MENU (item, nonpgm)	Edit matrix		[EDIT]	M.EDIT	Open matrix editor for matrix in X (Info : CAT.MENUS M_EDIT ; MIM = Matrix Input Mode ; [▲] or [▼] to move up or down one position ; Ref : 1/x, which is a shortcut for $[M]^{-1}$, use -1 y ^x instead ; and $\sqrt{x}$, which is a shortcut for MSQRT, use 0.5 y ^x instead) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_EDIT_MATX_page_1.pdf
	MENU (item, nonpgm)	Edit matrix (named)		[EDITN]	M.EDITN	Open matrix editor for matrix in register or variable (TAM : M.EDITN ... Tam menu) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_EDITN_page_1.pdf

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU	Matrix		[MATX]	MATX	Matrix functions ; a matrix is displayed in X and shown in other stack registers as [m×n Matrix] or [m×n C Matrix] when containing complex element(s) (Ref : https://47calc.com/doc/Application%20Notes/AN0020_C47_R47_Matrix_and_Vector.pdf ; Info : monadic functions work per matrix element (except 1/x, which is a shortcut for [M] ⁻¹ , use -1 y* instead ; and $\sqrt{x}$, which is a shortcut for MSQRT, use 0.5 y* instead) ; in edit mode, all functions work per element ; use registers to input calculation results) ; Keyboard f[9] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MATX_page_1.pdf
	MENU	Matrix editor		[M_EDIT]	M_EDIT	Matrix editor (Info : MIM = Matrix Input Mode ; [▲] or [▼] to move up or down one position ; Ref : 1/x, which is a shortcut for [M] ⁻¹ , use -1 y* instead ; and $\sqrt{x}$, which is a shortcut for MSQRT, use 0.5 y* instead) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_M_EDIT_page_1.pdf
	MENU	Matrix simultaneous equations		[M_SIMQ]	M_SIMQ	Matrix simultaneous equations : Mat A · Mat X = Mat B, where Mat A is the equation coefficient matrix and Mat B is the equation result vector ; Mat X is the equation solution vector Docs: https://47calc.com/doc/C47/Menus/C47_Menu_M_SIMQ_page_1.pdf
	Function (monadic)	Column or Manhattan norm		CNORM	CNORM	Maximum column sum of absolute element values of a matrix in X : $\ x\ _1 = \max_j \sum_i a_{ij} $ (Info : If CNORM is selected from menu [TamNorm.1] fF4, it enters $\ x\ _p$ with parameter 1 : Maximum column sum of absolute element values of a matrix in X : $\ x\ _1 = \max_j \sum_i a_{ij} $)
	Function (monadic)	Determinant		M	M	Determinant of matrix in X (matrix must be square)
	Function (monadic)	Dimension (flow)		DIM	M.DIM	Set dimension of matrix (named register or variable) to Y rows and X columns ; refills elements by row and cuts off excess elements or pads with zero elements ; creates a new matrix if necessary (TAM : M.DIM __ Tam menu ; Info : also see function DIMgr)
	Function (monadic)	Dimension (grow)		DIMgr	M.DIMgr	Set dimension of matrix (named register or variable) to Y rows and X columns ; cuts off rows and columns or pads with zero element rows and columns ; creates a new matrix if necessary (TAM : M.DIMgr __ Tam menu ; Info : also see function DIM)
	Function (monadic)	Eigenvalues		EIGVAL	EIGVAL	Eigenvalues as a 1xn vector (returned in X) and as the diagonal elements of an nxn matrix (returned in Y) ; tolerance set by SDIGS (use value of 10 ⁻¹⁶ for better performance or higher for better accuracy) (Info : Displays : i/m Tol: current/target Iter: n, where i/m is count of eigenvalues, n is iteration counter ; shown if flag MONIT is SET ; for new models and simulator, extended internal eigenvalue precision with 159-digit type integration was added, achieving 39-digit accuracy for 2x2 and 3x3 solvers, but this is not applicable to DM42 hardware)
	Function (monadic)	Eigenvector		EIGVEC	EIGVEC	Eigenvector of matrix in X ; tolerance set by SDIGS (use value of 10 ⁻¹⁶ for better performance or higher for better accuracy) (Info : Displays : i/m Tol: current/target Iter: n, where i/m is count of eigenvalues, n is iteration counter ; shown if flag MONIT is SET ; for new models and simulator, extended internal eigenvalue precision with 159-digit type integration was added, achieving 39-digit accuracy for 2x2 and 3x3 solvers, but this is not applicable to DM42 hardware)
	Function (monadic)	Euclidean or Frobenius norm		ENORM	ENORM	Euclidean magnitude of a vector in X or Frobenius norm of a matrix in X : $\ x\ _2 = \sqrt{\sum_i \sum_j a_{ij} ^2}$ (Info : If ENORM is selected from menu [TamNorm.1] fF5, it enters $\ x\ _p$ with parameter 2 : Euclidean magnitude of a vector in X or Frobenius norm of a matrix in X : $\ x\ _2 = \sqrt{\sum_i \sum_j a_{ij} ^2}$)
	Function (monadic)	Find element		FIND	M.FIND	Find element X of indexed matrix, setting row index I and column indexes if found (unchanged otherwise) (Info : If used in a program, the next step after M.FIND is performed if the element is found, otherwise it is skipped)
	Function (monadic)	Invert matrix		[M] ⁻¹	[M] ⁻¹	Invert matrix in X (exists only when M ≠ 0) (Shortcut : [1/x] ; Info : Error 22 -- Singular matrix -- is raised when M = 0)
	Function (monadic)	LU decomposition		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 (Info : LU (LR) factorisation, underlies inversion and linear solving)
	Function (monadic)	Maximum column value		CMAX	M.CMAX	Maximum column value for column (J) of indexed matrix, returns row index (I) in Y and maximum in X (Info : Set starting row and column index using STOIJ before executing CMAX)
	Function (monadic)	Minimum column value		CMIN	M.CMIN	Minimum column value for column (J) of indexed matrix, returns row index (I) in Y and minimum in X (Info : Set starting row and column index using STOIJ before executing CMIN)
	Function (monadic)	Norm		$\ x\ _p$	$\ x\ _p$	Norm of vector or matrix in X with parameter p (trailing input) = 0, 1, 2, ..., 9, ∞ : 0 : Number of non-zeros of vector or matrix in X : $\ x\ _0 = \sum_i \sum_j I(a_{ij} \neq 0)$, where I is the indicator function, 1 if true, 0 if false ; 1 : Maximum column sum of absolute element values of a matrix in X : $\ x\ _1 = \max_j \sum_i a_{ij} $; 2 : Euclidean magnitude of a vector in X or Frobenius norm of a matrix in X : $\ x\ _2 = \sqrt{\sum_i \sum_j a_{ij} ^2}$; 3 : $\ x\ _3 = \sqrt[3]{\sum_i \sum_j a_{ij} ^3}$; ... ; 9 : $\ x\ _9 = \sqrt[9]{\sum_i \sum_j a_{ij} ^9}$; ∞ : Maximum row sum of absolute element values of a matrix in X : $\ x\ _\infty = \max_i \sum_j a_{ij} $ (TAM : $\ x\ _p$ _ TamNorm menu ; Info : parameter 0, 1, 2, ..., 9, ∞ (use [TamNorm.1] F2 to enter ∞))
	Function (monadic)	Number of non-zeros		NNZ	NNZ	Number of non-zeros of vector or matrix in X : $\ x\ _0 = \sum_i \sum_j I(a_{ij} \neq 0)$, where I is the indicator function, 1 if true, 0 if false (Info : If NNZ is selected from menu [TamNorm.1] fF2, it enters $\ x\ _p$ with parameter 0 : Number of non-zeros of vector or matrix in X : $\ x\ _0 = \sum_i \sum_j I(a_{ij} \neq 0)$, where I is the indicator function, 1 if true, 0 if false)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic)	QR decomposition		M.QR	M.QR	Decompose matrix in X into unitary (Q) in Y and upper (R) matrix in X (Info : QR factorisation, underlies eigenvalue computation)
	Function (monadic)	Row or Maximum norm		RNORM	RNORM	Maximum row sum of absolute element values of a matrix in X : $\ x\ _{\infty} = \max_i \sum_j a_{ij} $ (Info : If RNORM is selected from menu [TamNorm.1] fF3, it enters $\ x\ _{\infty}$ with parameter ∞ : Maximum row sum of absolute element values of a matrix in X : $\ x\ _{\infty} = \max_i \sum_j a_{ij} $)
	Function (monadic)	Square root of matrix		MSQRT	M.SQRT	Square root of matrix in X : if $M = \text{MSQRT}(X)$, $M \times M = X$, where $\times$ denotes matrix multiplication (Shortcut : [$\sqrt{x}$] ; Info : Error 20 -- No root found or Error 21 -- Matrix mismatch may be raised (the latter when the matrix is not square))
	Function (monadic)	Transpose matrix		$[M]^T$	$[M]^T$	Transpose matrix in X
	Function (monadic, legacy)	Determinant		>DET<		Legacy version of function $ M $: Determinant of matrix in X (matrix must be square) (Info : Function >DET< is superceded by $ M $ on [MATX.2]F5)
	Function (monadic, legacy)	Invert matrix		>INVRT<		Legacy version of function $[M]^{-1}$: Invert matrix in X (exists only when $ M \neq 0$) (Info : Function >INVRT< is superceded by $[M]^{-1}$ on [MATX.2]F1)
	Function (monadic, legacy)	Transpose matrix		>TRANS<		Legacy version of function $[M]^T$: Transpose matrix in X (Info : Function >TRANS< is superceded by $[M]^T$ on [MATX.1]F2)
	Function (dyadic)	Vector angle		$\angle \vec{x}, \vec{y}$	$\angle \vec{x}, \vec{y}$	Angle between the 2D or 3D vectors in X and Y
	Function (42, monadic)	Matrix dimension (X)		42DIM#	42DIM#	Get dimensions of matrix in X ; this is a HP-42S/DM42(n) calculator compatible function
	Flag (system)	Wrap edge	CLEAR	WRPEDG		SET for wrap edge ; CLEAR for wrap edge inactive
	Flag (system)	Wrap end	CLEAR	WRPEND		SET for wrap end ; CLEAR for wrap end inactive
	Flag	Grow (matrix edit)	CLEAR	GROW		SET for grow (matrix edit) ; CLEAR for wrap (matrix edit) (Info : SBI is shown if flag SBmx is SET)
	Command (submnu)	New matrix		NEW	M.NEW	Create new matrix with Y rows and X columns
	Command (strike)	Concatenate (bottom)		CNCATB	M.CNCATB	Concatenate matrices (bottom) (TAM : M.CNCATB __ Tam menu)
	Command (strike)	Concatenate (right)		CNCATR	M.CNCATR	Concatenate matrices (right) (TAM : M.CNCATR __ Tam menu)
	Command (strike, old)	Concatenate (right)		>CNCAT<		Old version of the command CNCATR (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use CNCATR on [MATX.1]fF6 instead)
	Command (old)	Recall vector element		>RCLVEL<		Old version of the command RCLVEL (Info : May be found in older user programs, use RCLVEL on [MATX.1]gF4 instead)
	Command (old)	Store X into vector element	0	>STOVEL<		Old version of the command STOVEL (Info : May be found in older user programs, use STOVEL on [MATX.1]gF3 instead)
	Command	Add column		COL+1	M.COL+1	Add column to matrix, at the right edge
	Command	Add row		ROW+1	M.ROW+1	Add row to matrix, at the bottom edge
	Command	Column sum		CSUM	CSUM	Column sum of vector or matrix in X : $\text{CSUM} = \sum_i a_{ij}$
	Command	Copy elements to registers		$V \rightarrow R_{nn}$	$V \rightarrow R_{nn}$	Copy elements of matrix or vector in X to ascending range of registers, starting with register nn (TAM), where numbers of registers depends on size of matrix or vector (TAM : $V \rightarrow R_{nn}$ __ TamIndirect menu ; be careful when using the range of stack registers (100-107) ; Ref : https://47calc.com/doc/C47/Reference/C47_Registers.pdf)
	Command	Decrement column index	<col>	J-	J-	Decrement column index J of indexed matrix ; wraps automatically ; does not put value on the stack ; menu button J- on [MATX.3]F5 displays value as subscript (only if indexed matrix exists) (TI : $M[I_r J_c] = \langle \text{name} \rangle [\langle \text{row} \rangle \langle \text{col} \rangle]$)
	Command	Decrement row index	<row>	I-	I-	Decrement row index I of indexed matrix ; wraps automatically ; does not put value on the stack ; menu button I- on [MATX.3]F2 displays value as subscript (only if indexed matrix exists) (TI : $M[I_r J_c] = \langle \text{name} \rangle [\langle \text{row} \rangle \langle \text{col} \rangle]$)
	Command	Delete column		DELC	M.DELC	Delete column from matrix, at the cursor position
	Command	Delete row		DELR	M.DELR	Delete row from matrix, at the cursor position
	Command	Fill matrix from registers		$R_{nn} \rightarrow V$	$R_{nn} \rightarrow V$	Fill matrix or vector in X from ascending range of registers starting with register nn (TAM), where numbers of registers depends on size of matrix or vector (TAM : $R_{nn} \rightarrow V$ __ TamIndirect menu ; Ref : https://47calc.com/doc/C47/Reference/C47_Registers.pdf)
	Command	Get submatrix		GETM	M.GETM	Get submatrix with Y rows and X columns from the indexed matrix starting at current element
	Command	Go to matrix column		GOTO Column		Go to matrix column for GOTO (TAM : GOTO Column __ TamNonReg menu)
	Command	Go to matrix element		GOTO	M.GOTO	Go to matrix element using GOTO Row ; GOTO Column (TAM : GOTO Row ; GOTO Column)
	Command	Go to matrix row		GOTO Row		Go to matrix row for GOTO (TAM : GOTO Row __ TamNonReg menu)
	Command	Increment column index	<col>	J+	J+	Increment column index J of indexed matrix ; wraps automatically ; does not put value on the stack ; menu button J+ on [MATX.3]F6 displays value as subscript (only if indexed matrix exists) (TI : $M[I_r J_c] = \langle \text{name} \rangle [\langle \text{row} \rangle \langle \text{col} \rangle]$)
	Command	Increment row index	<row>	I+	I+	Increment row index I of indexed matrix ; wraps automatically ; does not put value on the stack ; menu button I+ on [MATX.3]F1 displays value as subscript (only if indexed matrix exists) (TI : $M[I_r J_c] = \langle \text{name} \rangle [\langle \text{row} \rangle \langle \text{col} \rangle]$)
	Command	Index matrix		INDEX	INDEX	Index matrix in a register or variable ; set row I to 1, column J to 1 ; clear flags WRPEDG and WRPEND ; index wraps automatically, setting these flags ; index is dropped upon exit of matrix editor (TAM : INDEX __ Tam menu ; TI : $M[I_r J_c] = \langle \text{name} \rangle [\langle \text{row} \rangle \langle \text{col} \rangle]$)
	Command	Insert column		INSC	M.INSC	Insert column into matrix, at the cursor position

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Insert row		INSR	M.INSR	Insert row into matrix, at the cursor position
	Command	Matrix dimension? (named)		DIMN#	M.DIMN#	Get dimensions of matrix in register or variable (TAM : M.DIMN# __ Tam menu)
	Command	Matrix X		Mat X	Mat_X	Create reserved Matrix variable for the Matrix X (Mat X) ; calculate equation solution vector using menu M_SIMQ ; created after RESET
	Command	New identity matrix		IDENT	M.IDENT	Create new identity matrix with Y rows and X columns
	Command	Old (matrix)		OLD	M.OLD	Revert to old element (while editing)
	Command	Put submatrix		PUTM	M.PUTM	Put submatrix in X into the indexed matrix starting at current element
	Command	Recall current matrix index		RCLIJ	RCLIJ	Recall current matrix index (row I and column J) (Shortcut : [RCL] - ...IJ ; TI : M<name> :I _r = ; M<name> :J _c =)
	Command	Recall matrix element		RCLEL	RCLEL	Recall current element of indexed matrix (Shortcut : [RCL] - ...EL ; TI : M[I _r J _c] = <name> [<row> <col>] =)
	Command	Recall matrix element and go to next		RCLSEQ	RCLSEQ	Recall current element of indexed matrix, then increment column index J of indexed matrix ; wraps automatically (TI : M[I _r J _c] = <name> [<row> <col>] =)
	Command	Recall vector element		RCLVEL	RCLVEL	Recall element n (TAM) of vector or matrix in X (Shortcut : [RCL] - ...VEL _{nn} ; TAM : RCLVEL ____ Tam menu)
	Command	Recall vector element 1		RCLVEL ₁	RCLVEL ₁	Recall element 1 of vector or matrix in X
	Command	Recall vector element 2		RCLVEL ₂	RCLVEL ₂	Recall element 2 of vector or matrix in X
	Command	Recall vector element 3		RCLVEL ₃	RCLVEL ₃	Recall element 3 of vector or matrix in X
	Command	Row sum		RSUM	RSUM	Row sum of vector or matrix in X : RSUM = $\sum a_{ij}$
	Command	Set current matrix index		ST0IJ	ST0IJ	Set current matrix index (row I and column J) (Shortcut : [ST0] - ...IJ ; TI : M<name> :I _r = ; M<name> :J _c =)
	Command	Store X into matrix element		ST0EL	ST0EL	Store X into current element of indexed matrix (Shortcut : [ST0] - Config ; TI : M[I _r J _c] = <name> [<row> <col>] =)
	Command	Store X into matrix element and go to next		ST0SEQ	ST0SEQ	Store X into current element of indexed matrix, then increment column index J of indexed matrix ; wraps automatically (TI : M[I _r J _c] = <name> [<row> <col>] =)
	Command	Store X into vector element		ST0VEL	ST0VEL	Store X into element n (TAM) of vector or matrix in Y (Shortcut : [ST0] - ...VEL _{nn} ; TAM : ST0VEL ____ Tam menu)
	Command	Store X into vector element 1		ST0VEL ₁	ST0VEL ₁	Store X into element 1 of vector or matrix in Y
	Command	Store X into vector element 2		ST0VEL ₂	ST0VEL ₂	Store X into element 2 of vector or matrix in Y
	Command	Store X into vector element 3		ST0VEL ₃	ST0VEL ₃	Store X into element 3 of vector or matrix in Y
	Command	Swap columns		C↔C	M.C↔C	Swap columns X and Y of indexed matrix
	Command	Swap rows		R↔R	M.R↔R	Swap rows X and Y of indexed matrix
Measurement	MENU (item)	Scatter plot		[SCATR]	SCATR	Scatter plot of measurements (programmable) (Info : CENTRL (re)starts menu SCATR ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_SCATR_page_1.pdf
	Command (submnu)	(Plot) Central		CENTRL	CENTRL	Orthogonal curve fit and show parameters, requires 30 data pairs (Info : Opens the graph mode and (re)starts menu SCATR)
	Command	Precision		S _{mi}	S _{mi}	Precision of measuring instrument investigated, requires 30 data pairs -- see https://blog.minitab.com/en/real-world-quality-improvement/orthogonal-regression-testing-the-equivalence-of-instruments (TI : S _{mi} =)
Mileage conversions	MENU	Mileage conversions		[Ymmv:]	Ymmv:	Convert between units of mileage ("Your mileage may vary") (Info : "E" designates "kWh" ; Flag CONV _H may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> ↔ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Ymmv_page_1.pdf
Miscellaneous conversions	MENU	Force loads conversions		[FLoads:]	FLoads:	Convert between units of force loads ; typical structural loads units (Info : Flag CONV _H may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> ↔ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FLoads_page_1.pdf
	MENU	Material properties conversions		[MatPrp:]	MatPrp:	Convert between units of material properties ; typical material properties given in material data sheets references (Info : Flag CONV _H may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> ↔ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MatPrp_page_1.pdf
	MENU	Miscellaneous conversions		[Misc:]	Misc:	Time, torque, power and field ratio conversions (Info : Flag CONV _H may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> ↔ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Misc_page_1.pdf
	MENU	Section properties conversions		[SecPrp:]	SecPrp:	Convert between units of section properties ; typical properties given in tables of steel/ALU sections (Info : Flag CONV _H may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> ↔ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_SecPrp_page_1.pdf

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU	Structural conversions		[Struct:]	Struct:	Section and material properties, force and moment load conversions (Info : Flag CONV _{HP} may be SET for <to> <-> <to> unit conversion buttons; CLEAR for <from> <-> <from> unit conversion buttons (default SET) ; Ref : https://47calc.com/doc/Application%20Notes/AN0027_C47_R47_DA_Structural_conversions.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Struct_page_1.pdf
Navigation	Command	Down		↓		Scroll down for menus and browser screens or single step through program steps (Info : To assign ↓ to a key or button, use ASN (f[1]) and spell the name as 'DOWN' ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard [▼]
	Command	Exit		EXIT		Exit (Info : To assign EXIT to a key or button, use ASN (f[1]) and spell the name as 'EXIT' ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard [EXIT]
	Command	Up		↑		Scroll up for menus and browser screens or backstep through program steps (Info : To assign ↑ to a key or button, use ASN (f[1]) and spell the name as 'UP' ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard [▲]
	Arrow	Move cursor left		←		Move cursor left
	Arrow	Move cursor right		→		Move cursor right
Navigation and characters	Arrow	Move down		↓		Move down one position (Fontcode : 2193 ₁₆)
	Arrow	Move left		←		Move left one position (Fontcode : 2190 ₁₆)
	Arrow	Move right		→		Move right one position (Fontcode : 2192 ₁₆)
	Arrow	Move up		↑		Move up one position (Fontcode : 2191 ₁₆)
Number bases	Setting (pgm)	Base number display	0	dBASE	dBASE ₀	Display long integer in X as base number in the T-line (base 2-62) ; exit by entering dBASE 0 (TAM : dBASE __ TamNonReg menu ; Split screen : Displaying X:z: <value> _{bb} ; Ref : https://47calc.com/doc/C47/Reference/C47_Base_digits.pdf)
	Setting (pgm)	Binary	OFF	BIN	BIN	Convert X to binary and toggle binary mode ; entering digits A-F switches the calculator to hexadecimal mode ; when KEY _{A-F} is SET, the respective digit is included in the row 2 key's longpress preview cycle (Shortcut : #TAM [1/x] (B) ; Info : values are displayed with subscript ₂)
	Setting (pgm)	Decimal (short integer)	OFF	DEC	DEC	Convert X to decimal (short integer) and toggle decimal (short integer) mode ; entering digits A-F switches the calculator to hexadecimal mode ; when KEY _{A-F} is SET, the respective digit is included in the row 2 key's longpress preview cycle (Shortcut : #TAM [LOG] (D) ; Info : values are displayed with subscript ₁₀)
	Setting (pgm)	Hexadecimal	OFF	HEX	HEX	Convert X to hexadecimal and toggle hexadecimal mode ; if KEY _{A-F} is SET, digits A-F can be entered by pressing row 2 keys (Shortcut : #TAM [RCL] (H) ; Info : values are displayed with subscript ₁₆)
	Setting (pgm)	Octal	OFF	OCT	OCT	Convert X to octal and toggle octal mode ; entering digits A-F switches the calculator to hexadecimal mode ; when KEY _{A-F} is SET, the respective digit is included in the row 2 key's longpress preview cycle (Shortcut : #TAM [E] (0) ; Info : values are displayed with subscript ₈)
	Setting (pgm)	Short integer split screen lines	2	dSI	dSI ₂	Set the short integer split screen lines for the display of X in #BASE Mode (TAM : dSI __ TamNonReg menu ; Info : Range is 0 : <none> ; 1 : HEX ; 2 : HEX/DEC ; 3 : HEX/BIN/DEC ; if flag BCD is SET, instead of HEX, BCD is displayed with subscript digits and followed by subscript _{bcd})
	Setting (pgm)	Signed 16 bits	OFF	16:2		Shortcut to set word size to 16 bits signed (ISM = 2) -- 2's complement (Info : SBI 16:2 is shown if flag SBint is SET)
	Setting (pgm)	Signed 32 bits	OFF	32:2		Shortcut to set word size to 32 bits signed (ISM = 2) -- 2's complement (Info : SBI 32:2 is shown if flag SBint is SET)
	Setting (pgm)	Signed 64 bits	ON	64:2		Shortcut to set word size to 64 bits signed (ISM = 2) -- 2's complement (Info : SBI 64:2 is shown if flag SBint is SET)
	Setting (pgm)	Signed 8 bits	OFF	8:2		Shortcut to set word size to 8 bits signed (ISM = 2) -- 2's complement (Info : SBI 8:2 is shown if flag SBint is SET)
	Setting (pgm)	Unsigned 16 bits	OFF	16:u		Shortcut to set word size to 16 bits unsigned (ISM = 0) -- Unsigned (Info : SBI 16:u is shown if flag SBint is SET)
	Setting (pgm)	Unsigned 32 bits	OFF	32:u		Shortcut to set word size to 32 bits unsigned (ISM = 0) -- Unsigned (Info : SBI 32:u is shown if flag SBint is SET)
	Setting (pgm)	Unsigned 64 bits	OFF	64:u		Shortcut to set word size to 64 bits unsigned (ISM = 0) -- Unsigned (Info : SBI 64:u is shown if flag SBint is SET)
	Setting (pgm)	Unsigned 8 bits	OFF	8:u		Shortcut to set word size to 8 bits unsigned (ISM = 0) -- Unsigned (Info : SBI 8:u is shown if flag SBint is SET)
	Setting (pgm)	Word size 16 bits	OFF	16-BIT	16-BIT ○	Set word size to 16 bits, for Short integer (8.02-8.16) (Info : <m> in SBI indicates sign mode (menu BITSET) ; SBI 16:<m> is shown if flag SBint is SET)
	Setting (pgm)	Word size 32 bits	OFF	32-BIT	32-BIT ○	Set word size to 32 bits, for Short integer (8.02-8.16) (Info : <m> in SBI indicates sign mode (menu BITSET) ; SBI 32:<m> is shown if flag SBint is SET)
	Setting (pgm)	Word size 64 bits	ON	64-BIT	64-BIT ⊗	Set word size to 64 bits, for Short integer (8.02-8.16) (Info : <m> in SBI indicates sign mode (menu BITSET) ; SBI 64:<m> is shown if flag SBint is SET)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Setting (pgm)	Word size 8 bits	OFF	8-BIT	8-BIT ☐	Set word size to 8 bits, for Short integer (8.02-8.16) (Info : <m> in SBI indicates sign mode (menu BITSET) ; SBI 8:<m> is shown if flag SBint is SET)
	Setting (pgm, stack)	Number base	OFF	#	→INT	Set number base ; operates on all stack registers depending on BASE _{HP} (TAM : →INT __ TamNonReg menu ; #TAM shortcuts : B = BIN ; D = DEC ; ENTER = DEC ; H = HEX ; 0 = OCT ; Info : SBI is shown if flag SBfrac is SET ; exit base mode by .d, →I, [,] or CLRMOD (longerpress [EXIT])) ; Keyboard g[1/x]
	Setting (pgm, stack)	Number base		→INT	→INT	Set number base ; operates on all stack registers depending on BASE _{HP} (TAM : →INT __ TamNonReg menu ; #TAM shortcuts : B = BIN ; D = DEC ; ENTER = DEC ; H = HEX ; 0 = OCT ; Info : SBI is shown if flag SBfrac is SET ; exit base mode by .d, →I, [,] or CLRMOD (longerpress [EXIT]))
	Setting	A-F keys	ON	KEY _{A-F}	KEY _{A-F} ☒	Switch on row 2 keys to enter digits A-F in #BASE Mode ; not programmable - in PEM, use flag KEY _{A-F} (Info : Entering digits A-F switches the calculator to hexadecimal mode ; when KEY _{A-F} is ON, the respective digit is included in the row 2 key's longpress preview cycle)
	Setting	Classic (HP) style base	ON	BASE _{HP}	BASE _{HP} ☒	Switch on applying number base to the whole stack ; not programmable - in PEM, use flag BASE _{HP} (Info : Shortint values only)
	MENU	Number base		[BASE]	BASE	Number base operations (Short integer (8.02-8.16)) (Split screen : Displaying value of X according to setting dSI (short integer split screen lines) : 0 : <none> ; 1 : HEX ; 2 : HEX/DEC ; 3 : HEX/BIN/DEC ; if flag BCD is SET, instead of HEX, BCD is displayed with subscript digits and followed by subscript _{bcd} ; Info : on BASE page 2, special indicators are used : ◇ : all group settings comply, ○ : ≥1 group settings differ) ; Keyboard f[4] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_BASE_page_1.pdf
	MENU	Number base (2)		[BASE2]	BASE2	Alternative menu for number base operations (Short integer (8.02-8.16)) (Split screen : Displaying value of X according to setting dSI (short integer split screen lines) : 0 : <none> ; 1 : HEX ; 2 : HEX/DEC ; 3 : HEX/BIN/DEC ; if flag BCD is SET, instead of HEX, BCD is displayed with subscript digits and followed by subscript _{bcd}) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_BASE2_page_1.pdf
	Function (monadic)	Byte swap		B.SWP	B.SWP	Swap bytes
	Function (monadic)	Clear bit		CB	CB	Clear bit n (TAM : CB __ TamNonReg menu)
	Function (monadic)	Get number of bits set		#B	#B	Get number of bits set
	Function (monadic)	Set bit		SB	SB	Set bit n (TAM : SB __ TamNonReg menu)
	Function (monadic)	Shift Left		SLn	SLn	Shift Left with number of bits (trailing input) (TAM : SLn __ TamNonReg menu)
	Function (monadic)	Shift Left (1)		SL	SL	Shortcut to Shift Left (1 bit)
	Function (monadic)	Shift Right		SRn	SRn	Shift Right with number of bits (trailing input) (TAM : SRn __ TamNonReg menu)
	Function (monadic)	Shift Right (1)		SR	SR	Shortcut to Shift Right (1 bit)
	Function (monadic)	Word swap		W.SWP	W.SWP	Swap words
	Flag	A-F keys	SET	KEY _{A-F}		SET for row 2 keys to enter digits A-F in #BASE Mode ; CLEAR for row 2 keys to work normally (Info : Entering digits A-F switches the calculator to hexadecimal mode ; when KEY _{A-F} is SET, the respective digit is included in the row 2 key's longpress preview cycle)
	Flag	Classic (HP) style base	SET	BASE _{HP}		SET for applying number base to the whole stack ; CLEAR for applying number base to X only (Info : Shortint values only)
	Command (logical)	Bit set?		BS?	BS?	Test bit n is set (TAM : BS? __ TamNonReg menu)
Number bases and characters	Symbol	Number sign		#		Number sign, used for Short integer (8.02-8.16) base (Shortcut : alpha f[.] ; Info : Fontcode : 0023 ₁₆)
Numeric input	Symbol	Radix		.		Enter radix (default ".") (Info : Second press enters fraction mode ; set default symbol in menu RADIX from menu [DISP.1]gF2) ; Keyboard [,]
	Setting (pgm)	Input Complex number	OFF	i ☒	i ☐	New numeric input is put on the stack as Complex number (2.0-2.5)
	Setting (pgm)	input Long integer	OFF	i LI	i LI ☐	New numeric input is put on the stack as Long integer (0) and entry of the decimal radix is ignored (Info : Switch on dZ.0 for displaying Long integers as Real ; not programmable - in PEM, use flag dZ.0)
	Setting (pgm)	Input Long integer/Real number	ON	i LI/R	i LI/R ☒	New numeric input is put on the stack as Long integer (0) or Real number (1.0-1.5) (Info : Uses the decimal radix (default ".") to determine the numeric datatype)
	Setting (pgm)	input Real number	OFF	i R	i R ☐	New numeric input is put on the stack as Real number (1.0-1.5)
	Setting	Binary prefixes	OFF	1024 ⁿ	1024 ⁿ ☐	Switch on binary prefixes in UNIT display mode ; not programmable - in PEM, use flag 1024 ⁿ (Info : Binary prefixes designate powers of 1024, applicable to long integers (will be indicated by Z:) and reals in UNIT display mode)
	Setting	Numlock	OFF	NUM		Switch on numlock ; not programmable - in PEM, use flag NUM (Shortcut : alpha f[▲] / alpha f[▼])
	MENU	Prefixes (UNIT)		[Prefix]	Prefix	Prefixes for UNIT mode Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Prefix_page_1.pdf
	Function (monadic)	Change sign		CHS	CHS	Change sign of number or time value in X (Info : CHS does not copy input to LASTx ; 0. CHS returns -0. provided flag SPCRES is SET) ; Keyboard [+/-]

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag	Binary prefixes	CLEAR	1024 ⁿ		SET for binary prefixes in UNIT display mode; CLEAR for decimal prefixes in UNIT display mode (Info : Binary prefixes designate powers of 1024, applicable to long integers (will be indicated by Z:) and reals in UNIT display mode)
	Flag	Numeric entry	CLEAR	NUM.IN		SET for numeric entry; CLEAR for numeric entry inactive
	Flag	Numlock	CLEAR	NUM		SET for numlock; CLEAR for numlock inactive
	Digit	Digit 0		0		Digit 0 (Fontcode : 0030 ₁₆) ; Keyboard [0]
	Digit	Digit 1		1		Digit 1 (Fontcode : 0031 ₁₆) ; Keyboard [1]
	Digit	Digit 2		2		Digit 2 (Fontcode : 0032 ₁₆) ; Keyboard [2]
	Digit	Digit 3		3		Digit 3 (Fontcode : 0033 ₁₆) ; Keyboard [3]
	Digit	Digit 4		4		Digit 4 (Fontcode : 0034 ₁₆) ; Keyboard [4]
	Digit	Digit 5		5		Digit 5 (Fontcode : 0035 ₁₆) ; Keyboard [5]
	Digit	Digit 6		6		Digit 6 (Fontcode : 0036 ₁₆) ; Keyboard [6]
	Digit	Digit 7		7		Digit 7 (Fontcode : 0037 ₁₆) ; Keyboard [7]
	Digit	Digit 8		8		Digit 8 (Fontcode : 0038 ₁₆) ; Keyboard [8]
	Digit	Digit 9		9		Digit 9 (Fontcode : 0039 ₁₆) ; Keyboard [9]
	Command (nonpgm)	femto		-f		Factor 10 ⁻¹⁵
	Command (nonpgm)	Giga		-G		Factor 10 ⁹
	Command (nonpgm)	Giga binary		-Gi		Factor 10 ³⁰
	Command (nonpgm)	kilo		-k		Factor 10 ³
	Command (nonpgm)	Kilo binary		-Ki		Factor 10 ¹⁰
	Command (nonpgm)	Mega		-M		Factor 10 ⁶
	Command (nonpgm)	Mega binary		-Mi		Factor 10 ²⁰
	Command (nonpgm)	micro		-μ		Factor 10 ⁻⁶
	Command (nonpgm)	milli		-m		Factor 10 ⁻³
	Command (nonpgm)	nano		-n		Factor 10 ⁻⁹
	Command (nonpgm)	Peta		-P		Factor 10 ¹⁵
	Command (nonpgm)	Peta binary		-Pi		Factor 10 ⁵⁰
	Command (nonpgm)	pico		-p		Factor 10 ⁻¹²
	Command (nonpgm)	Tera		-T		Factor 10 ¹²
	Command (nonpgm)	Tera binary		-Ti		Factor 10 ⁴⁰
	Command	Clear number		CLN		Clear numeric input buffer (Shortcut : NIM long [ε])
	Command	Enter exponent		EEK		Enter exponent (decimal input, powers of 10) ; Keyboard [E]
Numeric input and characters	Character	A		A		Character A (Fontcode : 0041 ₁₆) ; Keyboard alpha [Σ+]
	Character	B		B		Character B (Fontcode : 0042 ₁₆) ; Keyboard alpha [1/x]
	Character	C		C		Character C (Fontcode : 0043 ₁₆) ; Keyboard alpha [√x]
	Character	D		D		Character D (Fontcode : 0044 ₁₆) ; Keyboard alpha [LOG]
	Character	E		E		Character E (Fontcode : 0045 ₁₆) ; Keyboard alpha [LN]
	Character	F		F		Character F (Fontcode : 0046 ₁₆) ; Keyboard alpha [XEQ]
Plotting	Variable (real)	Lower limit for X-axis	-10	↓X	↓X	Reserved Real variable for the lower limit for X-axis (↓X) ; set interactively from menu Graphs and set by Draw _{xy} (TI : '↓X' :)
	Variable (real)	Lower limit for Y-axis	0	↓Y	↓Y	Reserved Real variable for the lower limit for Y-axis (↓Y) ; set interactively from menu Graphs and set by Plot functions (TI : '↓Y' :)
	Variable (real)	Upper limit for X-axis	10	↑X	↑X	Reserved Real variable for the upper limit for X-axis (↑X) ; set interactively from menu Graphs and set by Draw _{xy} (TI : '↑X' :)
	Variable (real)	Upper limit for Y-axis	10	↑Y	↑Y	Reserved Real variable for the upper limit for Y-axis (↑Y) ; set interactively from menu Graphs and set by Plot functions (TI : '↑Y' :)
	Setting (strike)	[future] Curve smoothing	OFF	CURVE		Switch on [future] curve smoothing ; not programmable - in PEM, use flag PLcurve (Info : Reserved for future use)
	Setting	Add differential curve	OFF	Δy/Δx		Switch on adding the graphical point-to-point differential of the main graph ; not programmable - in PEM, use flag PLdiff (Info : Δ Numerical slope is displayed ; switching ON ∫AREA also switches ON Δy/Δx ; switching OFF Δy/Δx also switches OFF ∫AREA ; cannot be combined with N.VECT, VECT)
	Setting	Add integral curve	OFF	ΣyΔx		Switch on adding the graphical integral curve, calculated from the point-to-point discrete integral ; not programmable - in PEM, use flag PLintg (Info : Vertical placement of the integral is at RMS of the graph points ; ∫ Trapezoid integral is displayed ; cannot be combined with N.VECT, VECT)
	Setting	Add RMS curve	OFF	RMS		Switch on adding the graphical root mean square curve, accumulated from left to right ; not programmable - in PEM, use flag PLrms (Info : ■ Root mean square RMS is displayed ; cannot be combined with N.VECT, VECT)
	Setting	Anti-clockwise vectors	OFF	VECT		Switch on anti-clockwise vectors ; not programmable - in PEM, use flag PLvec (Info : VECT and N.VECT are linked (both cannot be ON at the same time) ; cannot be combined with RMS, ∫AREA, Δy/Δx, ΣyΔx)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Setting	Autoscaling to show x-axis	OFF	X.AXIS		Switch on autoscaling to show x-axis ; not programmable - in PEM, use flag PLx.ax (Info : axis 0.0 or x-axis y=0 or y-axis x=0 is displayed)
	Setting	Autoscaling to show y-axis	OFF	Y.AXIS		Switch on autoscaling to show y-axis ; not programmable - in PEM, use flag PLy.ax (Info : axis 0.0 or x-axis y=0 or y-axis x=0 is displayed)
	Setting	Box markers	ON	BOX		Switch on box markers ; not programmable - in PEM, use flag PLbox (Info : If BOX is switched ON, CROSS and PLUS are switched OFF ; LINE is switched ON when all three curve markers are deselected)
	Setting	Clockwise vectors	OFF	N.VECT		Switch on clockwise vectors ; not programmable - in PEM, use flag PLnvec (Info : N.VECT and VECT are linked (both cannot be ON at the same time) ; cannot be combined with RMS, $\int$ AREA, $\Delta y/\Delta x$, $\Sigma \Delta x$)
	Setting	Complex plot	OFF	CXPLT		Switch on Complex plot ; not programmable - in PEM, use flag PLcxp (Info : CXPLT and IMPLT are linked (both cannot be ON at the same time))
	Setting	Cross markers	OFF	CROSS		Switch on cross markers ; not programmable - in PEM, use flag PLcros (Info : If CROSS is switched ON, BOX and PLUS are switched OFF ; LINE is switched ON when all three curve markers are deselected)
	Setting	Draw line	ON	LINE		Switch on connecting graph points using line segments ; not programmable - in PEM, use flag PLline (Info : if LINE is switched OFF, BOX may be switched ON if necessary (when no curve markers are selected))
	Setting	Forcing autoscale	OFF	X:Y=1		Switch on forcing autoscale on x- and y-axis to be the same ; not programmable - in PEM, use flag PLxy:1
	Setting	Plot imaginary part	OFF	IMPLT		Switch on plotting imaginary part ; not programmable - in PEM, use flag PLimp (Info : IMPLT and CXPLT are linked (both cannot be ON at the same time))
	Setting	Plus markers	OFF	PLUS		Switch on plus markers ; not programmable - in PEM, use flag PLplus (Info : If PLUS is switched ON, BOX and CROSS are switched OFF ; LINE is switched ON when all three curve markers are deselected)
	Setting	Shade curve	OFF	$\int$ AREA		Switch on shading the trapezoid integral from the graph points ; not programmable - in PEM, use flag PLshad (Info : $\int$ o Trapezoid integral is displayed ; switching ON $\int$ AREA also switches ON $\Delta y/\Delta x$; switching OFF $\Delta y/\Delta x$ also switches OFF $\int$ AREA ; cannot be combined with N.VECT, VECT)
	MENU (item)	Draw (stack limits)		[Draw _{xy}]	Draw _{xy}	Draw graph for expression (or equation) entered in the equation editor ; X-axis limits set from stack (Y, X) and Y-axis limits by variables (ΨY , ΨY) (Info : In Program Entry Mode, use PAUSE, SNAP or PLTFCNS to view the plot ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS ; if an equation was entered (containing "="), the left hand side is plotted) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DrawXY_page_1.pdf
	MENU (item)	Draw (variable limits)		[Draw]	Draw	Draw graph for expression (or equation) entered in the equation editor ; X-axis limits set by variables (ΨX , ΨX) and Y-axis limits by variables (ΨY , ΨY) (Info : In Program Entry Mode, use PAUSE, SNAP or PLTFCNS to view the plot ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS ; if an equation was entered (containing "="), the left hand side is plotted) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Draw_page_1.pdf
	MENU (item)	Plot functions		[PLTFCNS]	PLTFCNS	Plot functions (programmable), displaying graph, value/tick, lower left and upper right coordinate and additional info according to settings ; displays plot options menu (Info : Please note that this item is in development -- Error 59 : Undefined menu name -- may be raised ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PLTFCNS_page_1.pdf
	MENU	Graphing...		[Graphs]	Graphs	Equation graphing functions Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Graphs_page_1.pdf
	Flag	[future] Curve smoothing	CLEAR	PLcurve		SET for [future] curve smoothing ; CLEAR for [future] curve smoothing inactive (Info : Reserved for future use)
	Flag	Add differential curve	CLEAR	PLdiff		SET for adding the graphical point-to-point differential of the main graph ; CLEAR for not adding the differential curve (Info : Δ Numerical slope is displayed)
	Flag	Add integral curve	CLEAR	PLintg		SET for adding the graphical integral curve, calculated from the point-to-point discrete integral ; CLEAR for not adding the integral curve (Info : Vertical placement of the integral is at RMS of the graph points ; $\int$ o Trapezoid integral is displayed)
	Flag	Add RMS curve	CLEAR	PLrms		SET for adding the graphical root mean square curve, accumulated from left to right ; CLEAR for not adding the RMS curve (Info : $\blacksquare$ Root mean square RMS is displayed)
	Flag	Anti-clockwise vectors	CLEAR	PLvec		SET for anti-clockwise vectors ; CLEAR for anti-clockwise vectors inactive (Info : Treat coordinate pairs as vectors, anti-clockwise from the positive x-axis as reference ; PLvec and PLnvec are linked (both cannot be SET at the same time))

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag	Autoscaling to show x-axis	CLEAR	PLx.ax		SET for autoscaling to show x-axis; CLEAR for not autoscaling to show x-axis
	Flag	Autoscaling to show y-axis	CLEAR	PLy.ax		SET for autoscaling to show y-axis; CLEAR for not autoscaling to show y-axis
	Flag	Box markers	SET	PLbox		SET for box markers; CLEAR for box markers inactive (Info : If PLbox is SET, PLcros and PLplus are CLEARed ; PLLine is SET when all three curve markers are deselected)
	Flag	Clockwise vectors	CLEAR	PLnvec		SET for clockwise vectors; CLEAR for clockwise vectors inactive (Info : Treat coordinate pairs as navigation vectors, clockwise from the positive y-axis as reference ; PLnvec and PLvec are linked (both cannot be SET at the same time))
	Flag	Complex plot	CLEAR	PLcxp		SET for Complex plot; CLEAR for Complex plot inactive (Info : PLcxp and PLimp are linked (both cannot be SET at the same time))
	Flag	Cross markers	CLEAR	PLcros		SET for cross markers; CLEAR for cross markers inactive (Info : If PLcros is SET, PLbox and PLplus are CLEARed ; PLLine is SET when all three curve markers are deselected)
	Flag	Draw line	SET	PLline		SET for connecting graph points using line segments; CLEAR for not connecting graph points using line segments (Info : If PLLine is CLEARed, PLbox may be SET if necessary (when no curve markers are selected))
	Flag	Forcing autoscale	CLEAR	PLxy:1		SET for forcing autoscale on x- and y-axis to be the same; CLEAR for not forcing autoscale on x- and y-axis to be the same
	Flag	Plot imaginary part	CLEAR	PLimp		SET for plotting imaginary part; CLEAR for plotting real part (Info : PLimp and PLcxp are linked (both cannot be SET at the same time))
	Flag	Plus markers	CLEAR	PLplus		SET for plus markers; CLEAR for plus markers inactive (Info : If PLplus is SET, PLbox and PLcros are CLEARed ; PLLine is SET when all three curve markers are deselected)
	Flag	Shade curve	CLEAR	PLshad		SET for shading the trapezoid integral from the graph points; CLEAR for not shading the curve (Info : $\int$ Trapezoid integral is displayed)
	Flag	Show all SI prefixes	CLEAR	PFx.All		SET for showing full range of SI prefixes in UNIT display mode; CLEAR for showing limited range of SI prefixes in UNIT display mode (Info : Full range is 10^{-30} to 10^{30} ; limited range is 10^{-15} to 10^{15} ; not applicable when flag 102 nd is SET ; Ref : https://47calc.com/doc/C47/Reference/C47_UNIT_prefix.pdf)
	Command (PEM)	Reset plot		PLTRST	PLTRST	Reset all plot options and redraw graph (Info : Resets options for lines, boxes, crosses, plusses and scaling ; PLTRST also resets $\downarrow Y$ and $\uparrow Y$ to zero)
	Command	List graph coordinates		LISTXY		List the actual STATS graph coordinates (7 digit floating point)
	Command	Zoom in on y-axis	0	+ZOOM		Increase zoom level on the y-axis (range -16 to +16) (Info : <0 : zoom out ; 0 : normal ; >0 : zoom in ; +ZOOM resets $\uparrow Y$ and $\uparrow Y$ to zero)
	Command	Zoom out on y-axis	0	-ZOOM		Decrease zoom level on the y-axis (range -16 to +16) (Info : <0 : zoom out ; 0 : normal ; >0 : zoom in ; -ZOOM resets $\uparrow Y$ and $\uparrow Y$ to zero)
Plotting and curve fitting	Setting (old)	Set Best Fit	1	>BestF<		Old version of the setting BestF that could not use indirect addressing (Info : May be found in older user programs, use BestF on [MODEL.1]gF3 instead)
	Setting	Cauchy peak fit	OFF	CauchF	CauchF ☐	Set <BestF> to include Cauchy peak curve fitting : $y = [a_0 (x + a_1)^2 + a_2]^{-1}$; deselects OrthoF (Info : Adds 128 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Exponential fit	OFF	ExpF	ExpF ☐	Set <BestF> to include Exponential curve fitting : $y = a_0 e^{(a_1 x)}$; deselects OrthoF (Info : Adds 2 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Gauss peak fit	OFF	GaussF	GaussF ☐	Set <BestF> to include Gauss peak curve fitting : $y = a_0 e^{[(x - a_1)^2 / a_2]}$; deselects OrthoF (Info : Adds 256 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Hyperbolic fit	OFF	HypF	HypF ☐	Set <BestF> to include Hyperbolic curve fitting : $y = (a_0 + a_1 x)^{-1}$; deselects OrthoF (Info : Adds 32 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Linear fit	ON	LinF	LinF ☒	Set <BestF> to include Linear curve fitting : $y = a_0 + a_1 x$; deselects OrthoF (Info : Adds 1 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Logarithmic fit	OFF	LogF	LogF ☐	Set <BestF> to include Logarithmic curve fitting : $y = a_0 + a_1 \ln(x)$; deselects OrthoF (Info : Adds 4 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Orthogonal fit	OFF	OrthoF	OrthoF ☐	Select orthogonal curve fitting (used in PLSTAT) ; deselects other curve fit settings (Info : Sets <BestF> to 0 ; TI : L.R. selected to OrthoF)
	Setting	Parabolic fit	OFF	ParabF	ParabF ☐	Set <BestF> to include Parabolic curve fitting : $y = a_0 + a_1 x + a_2 x^2$; deselects OrthoF (Info : Adds 64 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Power fit	OFF	PowerF	PowerF ☐	Set <BestF> to include Power curve fitting : $y = a_0 x^{a_1}$; deselects OrthoF (Info : Adds 8 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Root fit	OFF	RootF	RootF ☐	Set <BestF> to include Root curve fitting : $y = a_0 a_1^{(1/x)}$; deselects OrthoF (Info : Adds 16 to <BestF> ; TI : L.R. selected to <BestF>.)
	Setting	Set Best Fit	1	BestF	BestF ₁	Select curve fit model(s) by setting parameter value directly (TAM : BestF ____ TamNonReg menu)
	MENU (item)	Assess		[ASSESS]		Display graphical assessments of the selected regression curves Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ASSESS_page_1.pdf
	MENU	Model		[MODEL]	MODEL	Functions to set parameter for which models are selected in the LR analysis (Ref : https://47calc.com/doc/C47/Reference/C47_BestF_param.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MODEL_page_1.pdf

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU	Regression		[REGR]	REGR	Regression functions Docs: https://47calc.com/doc/C47/Menus/C47_Menu_REGR_page_1.pdf
	Command (cyclic, nonpgm)	Next fit		NXTFIT		Assess next curve fitting (cyclic)
	Command	Fit model correlation		r	CORR	Fit model correlation
	Command	Fit model covariance		cov	COV	Fit model covariance
	Command	Get Best Fit		BestF#	BestF#	Get parameter for which models are selected in the LR analysis (Info : Set in menu MODEL)
	Command	Linear Regression		L.R.	L.R.	Calculate all selected regression models, and select the one with the best coefficient of fit model correlation (r) (TI : <model>* a ₂ = ; y = a ₁ = ; <formula> a ₀ =)
	Command	Reset fitting		ResetF	ResetF	Reset <BestF> to default (1) (TI : L.R. selected to 001)
	Command	Retrieve LR-param a ₀		LR:a0	LR:a0	Retrieve linear regression parameter a ₀ and show formula and parameter name (TI : y = <formula> : a ₀ =)
	Command	Retrieve LR-param a ₁		LR:a1	LR:a1	Retrieve linear regression parameter a ₁ and show formula and parameter name (TI : y = <formula> : a ₁ =)
	Command	Retrieve LR-param a ₂		LR:a2	LR:a2	Retrieve linear regression parameter a ₂ and show formula and parameter name (TI : y = <formula> : a ₂ =)
	Command	s _{xy}		s _{xy}	s _{xy}	Sample covariance
	Command	Set all (models)		AllF	AllF	Set <BestF> to include all curve fitting models (511) ; deselects OrthoF (TI : L.R. selected to 511)
	Command	Standard errors		s(a)	s(a)	Standard errors of line fitted ; s(a ₀) in X ; s(a ₁) in Y (Info : Works for ExpF ; LinF ; LogF ; OrthoF ; PowerF ; does not work for CauchF ; GaussF ; HypF ; ParabF ; RootF)
	Command	x hat		$\hat{x}$	$\hat{x}$	Estimation of x
	Command	y hat		$\hat{y}$	$\hat{y}$	Estimation of y
Plotting and statistics	Variable (matrix)	Histogram matrix		HIST0	HIST0	Reserved Matrix variable (cannot be deleted) for the histogram matrix (HIST0) ; created from STATS using menu HIST ; not created after RESET
	Variable (matrix)	Statistics matrix		STATS	STATS	Reserved Matrix variable (cannot be deleted) for the statistics matrix (STATS) ; created from data entered using Σ^+ ; not created after RESET (TI : Re-calculating sums ... ; may be shown when storing or loading the statistics matrix)
	MENU (item)	Histogram		[HPLOT]	HPlot	Histogram plotting (programmable) (Info : CAT.MENUS HPlot) Docs: https://47calc.com/doc/C47/Menus/xxx1.pdf
	MENU (item)	Plot statistics		[PLSTAT]	PLSTAT	Plot statistics matrix (STATS, programmable) (TI : Calculating graph coordinates ... may be shown when plotting the statistics matrix ; Info : in Program Entry Mode, use PAUSE, SNAP or PLTFCNS to view the plot (PLTFCNS also displays the plot options menu) ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS ; Ref : https://47calc.com/doc/Application%20Notes/AN0003_C47_JM_Graphing.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PLSTAT_page_1.pdf
	MENU	Histogram		[HIST]	HIST	Histogram functions Docs: https://47calc.com/doc/C47/Menus/C47_Menu_HIST_page_1.pdf
	MENU	Histogram		[HPLOT]	HPlot	Histogram plotting (programmable) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_HPlot_page_1.pdf
	Command	High bin		↑BIN	↑BIN	High bin (TI : ↑BIN : ; ↓BIN : ; nBINS :)
	Command	Histogram Normal		HNORM	HNORM	Fit Gauss distribution through HIST0 data (programmable)
	Command	Histogram X		HISTOX	HISTOX	Evaluate first column of STATS and store in HIST0 (TI : ↑BIN : ; ↓BIN : ; nBINS :)
	Command	Histogram Y		HISTOY	HISTOY	Evaluate second column of STATS and store in HIST0 (TI : ↑BIN : ; ↓BIN : ; nBINS :)
	Command	Low bin		↓BIN	↓BIN	Low bin (TI : ↑BIN : ; ↓BIN : ; nBINS :)
	Command	Number of bins		nBINS		Number of bins in histogram (TI : ↑BIN : ; ↓BIN : ; nBINS :)
Plotting and summation	MENU	Plotting		[PLOT]	PLOT	Plotting and summation functions (Info : When SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS) ; Keyboard g[:] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PLOT_page_1.pdf
Power conversions	MENU	Power conversions		[Power:]	Power:	Convert between units of power (Info : Flag CONV _{HP} may be SET for <to> $\leftrightarrow$ <to> unit conversion buttons ; CLEAR for <from> $\rightarrow$ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Power_page_1.pdf
	Function (monadic, linked)	hp _E to W		hp _E →W		Convert electrical horsepower to Watt (TI : W _E)
	Function (monadic, linked)	hp _M to W		hp _M →W		Convert metric horsepower to Watt (TI : W _M)
	Function (monadic, linked)	hp _{UK} to W		hp _{UK} →W		Convert UK horsepower to Watt (TI : W _U)
	Function (monadic, linked)	W to hp _E		W→hp _E		Convert Watt to electrical horsepower (TI : hp _E)
	Function (monadic, linked)	W to hp _M		W→hp _M		Convert Watt to metric horsepower (TI : hp _M)
	Function (monadic, linked)	W to hp _{UK}		W→hp _{UK}		Convert Watt to UK horsepower (TI : hp _{UK})
Pressure conversions	MENU	Pressure loads conversions		[PLoads:]	PLoads:	Convert between units of pressure loads ; typical loads presented as pressures (Info : Flag CONV _{HP} may be SET for <to> $\leftrightarrow$ <to> unit conversion buttons ; CLEAR for <from> $\rightarrow$ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PLoads_page_1.pdf
	Function (monadic, linked)	atm to Pa		atm→Pa		Convert atmosphere to Pascal (TI : Pa)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, linked)	bar to Pa		bar→Pa		Convert bar to Pascal (TI : Pa:)
	Function (monadic, linked)	inHg to Pa		inHg→Pa		Convert inch of Mercury to Pascal (TI : Pa:)
	Function (monadic, linked)	ksi to MPa		ksi→MPa		Convert kilo-pound force per square inch to MegaPascal (TI : MPa:)
	Function (monadic, linked)	lbf/ft ² to Pa		lbf/ft ² →Pa		Convert pound force per square foot to Pascal (TI : Pa:)
	Function (monadic, linked)	mmHg to Pa		mmHg→Pa		Convert millimetre of Mercury to Pascal (TI : Pa:)
	Function (monadic, linked)	MPa to ksi		MPa→ksi		Convert MegaPascal to kilo-pound force per square inch (TI : ksi:)
	Function (monadic, linked)	Pa to atm		Pa→atm		Convert Pascal to atmosphere (TI : atm:)
	Function (monadic, linked)	Pa to bar		Pa→bar		Convert Pascal to bar (TI : bar:)
	Function (monadic, linked)	Pa to inHg		Pa→inHg		Convert Pascal to inch of Mercury (TI : inHg:)
	Function (monadic, linked)	Pa to lbf/ft ²		Pa→lbf/ft ²		Convert Pascal to pound force per square foot (TI : lbf/ft ² :)
	Function (monadic, linked)	Pa to mmHg		Pa→mmHg		Convert Pascal to millimetre of Mercury (TI : mmHg:)
	Function (monadic, linked)	Pa to psi		Pa→psi		Convert Pascal to pound force per square inch (TI : psi:)
	Function (monadic, linked)	Pa to torr		Pa→torr		Convert Pascal to torr (TI : torr:)
	Function (monadic, linked)	psi to Pa		psi→Pa		Convert pound force per square inch to Pascal (TI : Pa:)
	Function (monadic, linked)	torr to Pa		torr→Pa		Convert torr to Pascal (TI : Pa:)
Printing	Setting (pgm)	Printer mode Manual	ON	MAN	MAN ☉	Switch on printer mode Manual (flag ☐TRACE is CLEAR and flag ☐NORM is CLEAR) (Info : In MAN mode, the printer is idle and only prints when a print command is executed ; VIEW and AVIEW can also generate print output in this mode)
	Setting (pgm)	Printer mode Normal	OFF	NORM	NORM ○	Switch on printer mode Normal (flag ☐TRACE is CLEAR and flag ☐NORM is SET) (Info : In NORM mode, the printer prints numbers and alpha strings that are keyed in, names of functions executed from the keyboard and output from print functions ; running programs only print from print commands or from PROMPT)
	Setting (pgm)	Printer mode Stack trace	OFF	STRACE	STRACE ○	Switch on printer mode Stack trace (flag ☐TRACE is SET and flag ☐NORM is SET) (Info : In STRACE mode, the printer prints the stack and alpha strings that are keyed in, names of functions executed from the keyboard, intermediate and final results and output from print functions ; if a program is running, each step is printed as it executes ; in this mode, program listings are printed in condensed form)
	Setting (pgm)	Printer mode Trace	OFF	TRACE	TRACE ○	Switch on printer mode Trace (flag ☐TRACE is SET and flag ☐NORM is CLEAR) (Info : In TRACE mode, the printer prints numbers and alpha strings that are keyed in, names of functions executed from the keyboard, intermediate and final results and output from print functions ; if a program is running, each step is printed as it executes ; in this mode, program listings are printed in condensed form)
	Setting (pgm)	Printing OFF	☐OFF	☐OFF	☐OFF ☉	Switch off printing and print to file (selected commands) (Info : ☐OFF clears flag ☐ACT (infrared print port inactive) and ☐EN (printing disabled) ; Error 63 -- Printing Is Disabled -- may be shown when executing a print command)
	Setting (pgm)	Printing ON	OFF	☐ON	☐ON ○	Switch on printing and send commands to infrared print port (Info : ☐ON sets flag ☐ACT (infrared print port active) and ☐EN (printing enabled))
	Setting (pgm)	Set printer delay	18	☐DELAY	☐DELAY 18	Set printer delay time between lines (TAM : ☐DELAY _ Tam menu ; Info : Set printer delay default (18) is set by DMCP ; on the simulator, the default is 0)
	Setting (pgm, strike)	Printer mode	0	☐MODE	☐MODE	Set printer mode (TAM : ☐MODE _ Tam menu ; Info : only mode 0 is implemented for the time being)
	Setting	HP-82240 printer	ON	82240	82240 ☉	Set the printer type to HP-82240 (Info : Select this option for the HP-82240A or HP-82240B, or any printer compatible with the HP-82240)
	Setting	Martel printer	OFF	Martel	Martel ○	Set the printer type to Martel (Info : The Martel printer supporting the HPIR interface may show some slight variations in character size or text position, in comparison to the 82240)
	Setting	Print port active	OFF	PRNTR	PRNTR □	Switch on infrared print port active ; not programmable
	MENU	Printer control	82240	[☐]	☐	Printer control ; the menu button, e.g. on [PRINT.1]gF1 shows the default printer selected : ☐82240 (Info : Printing is available for new models and simulator, but is currently available on DM42 hardware only in some packages) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Printer_page_1.pdf
	MENU	Printing		[PRINT]	PRINT	Printing functions (Info : Printing is available for new models and simulator, but is currently available on DM42 hardware only in some packages ; Ref : https://47calc.com/doc/Application%20Notes/AN0021_C47_R47_DL_Printing.pdf) ; Keyboard f[-] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PRINT_page_1.pdf
	Flag (system)	Print port active	CLEAR	☐ACT		SET for infrared print port active ; CLEAR for infrared print port inactive (Info : If flag ☐ACT is SET, the infrared port is enabled ; if CLEAR, the infrared port is disabled)
	Flag	Printer mode Normal/Trace	CLEAR	☐NORM		SET for printer mode Normal (flag ☐TRACE is CLEAR) or printer mode Stack trace (flag ☐TRACE is SET) ; CLEAR for printer mode Manual (flag ☐TRACE is CLEAR) or printer mode Trace (flag ☐TRACE is SET) (Info : There are four print modes : MAN, NORM, TRACE and STRACE, which are set by two flags : ☐TRACE and ☐NORM)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag	Printer mode Trace/Stack trace	CLEAR	TRACE		SET for printer mode Trace (flag NORM is CLEAR) or printer mode Stack trace (flag NORM is SET); CLEAR for printer mode Manual (flag NORM is CLEAR) or printer mode Normal (flag NORM is SET) (Info : There are four print modes : MAN, NORM, TRACE and STRACE, which are set by two flags : TRACE and NORM)
	Flag	Printing enabled	CLEAR	EN		SET for printing enabled; CLEAR for printing disabled
	Command (strike)	Print LCD		LCD	LCD	Print LCD display ; on the Martel, the full LCD can be printed, but on the 82240 this has not been implemented yet ; note that printing graphics is slow (Info : If OFF, LCD works like SNAP : writes LCD screenshot to file SCREENS/<YYYYMMDD-HHMMSS00>.bmp and writes stack to file DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)
	Command (strike)	Print XFN		XFN	XFN	Print all digits of X × Y + Z for Virtual compound data type (0/1.0-1.5 ; 0/1.0 ; 0/1.0), used in extended accuracy functions (menu XFCNS) (Info : If OFF, writes XFN to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)
	Command (strike)	Printer width		WIDTH	WIDTH	Number of columns used to print a string specified in a register or variable (Info : The number of columns returned depends on the printer type selected and (possibly) on MODE ; SBI is shown if flag SBprn is SET)
	Command (SIM)	Print all items		ITEMS		Print all items (for testing purposes on simulator only, not available on hardware) (Info : [EXIT] or [R/S] can be pressed at any time to stop the listing ; If OFF, writes all items to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)
	Command (old)	Print byte		>#<		Old version of the command # that used a different programming parameter (Info : May be found in older user programs, use # on [PRINT.1] fF3 instead)
	Command (old)	Print character		>CHAR<		Old version of the command CHAR that used a different programming parameter (Info : May be found in older user programs, use CHAR on [PRINT.1] fF5 instead)
	Command (old)	Print column		>TAB<		Old version of the command TAB that used a different programming parameter (Info : May be found in older user programs, use TAB on [PRINT.2] F5 instead)
	Command (nonpgm)	Print list		LIST	LIST	Print specified number of lines of current program (TAM : LIST ____ TamNonReg menu ; Info : [EXIT] or [R/S] can be pressed at any time to stop the listing ; in printer mode Trace, the program listing is printed in condensed form ; SBI is shown if flag SBprn is SET)
	Command (nonpgm)	Print program		PROG	PROG	Print program, with date and time printed at the top of the listing (TAM : PROG _ TamLb1Only menu ; Info : If OFF, executes XPORTP ; [EXIT] or [R/S] can be pressed at any time to stop the listing ; in printer mode Trace, the program listing is printed in condensed form ; SBI is shown if flag SBprn is SET)
	Command	Get printer delay		DELAY#		Get printer delay
	Command	Print all registers		ALLr	ALLr	Print all registers, labeled ; prints a listing of all global and currently allocated local registers (if any) (Info : [EXIT] or [R/S] can be pressed at any time to stop the listing ; If OFF, writes all registers to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)
	Command	Print alpha		α	α	Print contents of register or variable as text (Info : If OFF, α writes alpha to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)
	Command	Print byte		#	#	Print single byte (value 0-255), to be used as character or ESC sequence (TAM : # ____ TamNonReg menu ; Info : SBI is shown if flag SBprn is SET)
	Command	Print character		CHAR	CHAR	Print single character from Unicode value specified in a register or variable (TAM : CHAR ____ Tam menu ; Info : only characters in the C47 standard font are supported ; SBI is shown if flag SBprn is SET)
	Command	Print column		TAB	TAB	Position print head to print column nnn (HP-82240 : 1-166 columns, Martel : 1-384 columns) (TAM : TAB ____ Tam menu ; Info : SBI is shown if flag SBprn is SET)
	Command	Print register or variable		r	r	Print single register or variable, labeled with the register or variable name (TAM : r __ Tam menu ; Info : if OFF, writes register or variable to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)
	Command	Print registers		REGS	REGS	Print registers, using range number in X (sss.nn means registers from sss through sss + nn - 1) (Info : [EXIT] or [R/S] can be pressed at any time to stop the listing ; If OFF, writes registers to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)
	Command	Print stack		STK	STK	Print stack registers, labeled, in TZYX or DCBA TZYX order, depending on stack size set by SSIZE8 (Info : If OFF, writes stack to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI ; 8 ; 4 is shown if flag SBprn is SET)
	Command	Print summation registers		Σ	Σ	Print summation registers, labeled (Info : SBI is shown if flag SBprn is SET)
	Command	Print user items		USER	USER	Print user items : variable names and values and a list of global program labels and program numbers (Info : [EXIT] or [R/S] can be pressed at any time to stop the listing ; SBI is shown if flag SBprn is SET)
	Command	Print X		x	x	Print X register, marked with *** (Info : If OFF, writes X to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI is shown if flag SBprn is SET)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Print X and Y				Print X and Y registers on the same line, X on the left, Y on the right ; a 2xn or nx2 matrix is printed in two columns ; maximum 11 characters per column including number sign and separator(s) ; this is very limited for Complex numbers (Info : If  , writes X and Y to file in DATA/<YYYYMMDD-HHMMSS00>.REGS.TSV ; SBI  is shown if flag SBprn is SET)
	Command	Printer advance				Advance print paper, after printing the contents of the print buffer (Info : SBI  is shown if flag SBprn is SET)
Probability	MENU	Distributions		[DISTR]	DISTR	Probability distribution functions (Split screen : Displaying <register> = <parameter> = <value> for the probability distribution submenus ; Ref : https://47calc.com/doc/C47/Reference/C47_Distr_param.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DISTR_page_1.pdf
	MENU	Probability		[PROB]	PROB	Probability functions ; Keyboard f[x] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PROB_page_1.pdf
	Function (dyadic)	Permutations		P _{yx}	PERM	Permutations of X out of Y
	Function (dyadic)	Random Long integer or Real		RANI#	RANI#	Pseudo random Long integer or Real number generator with bounds X and Y ; uses the SEED (Info : Result is Long integer and range is [X, Y] when X and Y are Long integer ; result is Real and range is [X, Y] when X or Y are Real ; bounds may be entered in either order)
	Command	Random number		RAN#	RAN#	Pseudo random number generator (type Real) with bounds 0 and 1 ; uses the SEED (Info : Range is [0, 1) -- left closed, right open interval)
	Command	Seed		SEED	SEED	Set seed for the pseudo random number generators RAN# and RANI# (Info : If X = 0, then SEED will use a combination of available memory and current time to create a random-like new seed ; for any other (numeric) input the resulting pseudo random sequence will be repeatable)
Probability (int)	MENU	Binomial & Negative binomial distribution		[Binom:]	Binom:	Binomial & Negative binomial probability distribution (discrete) (TI : RegP = p = ; RegN = n =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Binom_page_1.pdf
	MENU	Discrete uniform distribution		[DisUni:]	DisUni:	Discrete uniform probability distribution (discrete) (TI : RegM = a = ; RegN = b =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DisUni_page_1.pdf
	MENU	Geometric distribution		[Geom:]	Geom:	Geometric probability distribution (discrete) (TI : RegP = p =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Geom_page_1.pdf
	MENU	Hypergeometric distribution		[Hyper:]	Hyper:	Hypergeometric probability distribution (discrete) (TI : RegM = N = ; RegN = n = ; RegQ = K =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Hyper_page_1.pdf
	MENU	Poisson distribution		[Poiss:]	Poiss:	Poisson probability distribution (discrete) -- see https://en.wikipedia.org/wiki/Poisson_distribution#Definitions (TI : RegR = λ =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Poiss_page_1.pdf
	Function (monadic)	Binomial cdf (lower)		Binom _L	Binom _L	Binomial cumulative distribution (lower tail)
	Function (monadic)	Binomial cdf (upper)		Binom _U	Binom _U	Binomial cumulative distribution (upper tail)
	Function (monadic)	Binomial pmf		Binom _P	Binom _P	Binomial probability mass function
	Function (monadic)	Binomial qf (inverse)		Binom ⁻¹	Binom ⁻¹	Binomial inverse cumulative distribution (quantile)
	Function (monadic)	Discrete uniform cdf (lower)		DisU _L	DisU _L	Discrete uniform cumulative distribution (lower tail)
	Function (monadic)	Discrete uniform cdf (upper)		DisU _U	DisU _U	Discrete uniform cumulative distribution (upper tail)
	Function (monadic)	Discrete uniform pmf		DisU _P	DisU _P	Discrete uniform probability mass function
	Function (monadic)	Discrete uniform qf (inverse)		DisU ⁻¹	DisU ⁻¹	Discrete uniform inverse cumulative distribution (quantile)
	Function (monadic)	Geometric cdf (lower)		Geom _L	Geom _L	Geometric cumulative distribution (lower tail)
	Function (monadic)	Geometric cdf (upper)		Geom _U	Geom _U	Geometric cumulative distribution (upper tail)
	Function (monadic)	Geometric pmf		Geom _P	Geom _P	Geometric probability mass function
	Function (monadic)	Geometric qf (inverse)		Geom ⁻¹	Geom ⁻¹	Geometric inverse cumulative distribution (quantile)
	Function (monadic)	Hypergeometric cdf (lower)		Hyper _L	Hyper _L	Hypergeometric cumulative distribution (lower tail)
	Function (monadic)	Hypergeometric cdf (upper)		Hyper _U	Hyper _U	Hypergeometric cumulative distribution (upper tail)
	Function (monadic)	Hypergeometric pmf		Hyper _P	Hyper _P	Hypergeometric probability mass function
	Function (monadic)	Hypergeometric qf (inverse)		Hyper ⁻¹	Hyper ⁻¹	Hypergeometric inverse cumulative distribution (quantile)
	Function (monadic)	Negative binomial cdf (lower)		NBin _L	NBin _L	Negative binomial cumulative distribution (lower tail)
	Function (monadic)	Negative binomial cdf (upper)		NBin _U	NBin _U	Negative binomial cumulative distribution (upper tail)
	Function (monadic)	Negative binomial pmf		NBin _P	NBin _P	Negative binomial probability mass function
	Function (monadic)	Negative binomial qf (inverse)		NBin ⁻¹	NBin ⁻¹	Negative binomial inverse cumulative distribution (quantile)
	Function (monadic)	Poisson cdf (lower)		Poiss _L	Poiss _L	Poisson cumulative distribution (lower tail)
	Function (monadic)	Poisson cdf (upper)		Poiss _U	Poiss _U	Poisson cumulative distribution (upper tail)
	Function (monadic)	Poisson pmf		Poiss _P	Poiss _P	Poisson probability mass function
	Function (monadic)	Poisson qf (inverse)		Poiss ⁻¹	Poiss ⁻¹	Poisson inverse cumulative distribution (quantile)
	Command	Store a in M		STO _M a		Store parameter a : lower limit in register M (Split screen : RegM = a =)
	Command	Store b in N		STO _N b		Store parameter b : upper limit in register N (Split screen : RegN = b =)
	Command	Store K in Q		STO _Q K		Store parameter K : number of special items in the pool in register Q (Split screen : RegQ = K =)
	Command	Store N in M		STO _M N		Store parameter N : population size in register M (Split screen : RegM = N =)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Store n in N		ST0N n		Store parameter n : number of samples in register N (Split screen : RegN = n =)
	Command	Store p in P		ST0p p		Store parameter p : probability in register P (Split screen : RegP = p =)
Probability (real)	MENU	Cauchy-Lorentz distribution		[Cauch:]	Cauch:	Cauchy-Lorentz probability distribution (continuous) (TI : RegM = x_0 = ; RegS = γ =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Cauch_page_1.pdf
	MENU	Chi-squared distribution		[χ^2 :]	χ^2 :	Chi-squared probability distribution (continuous) (TI : RegM = ν =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Chi2_page_1.pdf
	MENU	Exponential distribution		[Expon:]	Expon:	Exponential probability distribution (continuous) (TI : RegR = λ =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Expon_page_1.pdf
	MENU	Fisher's F distribution		[F:]	F:	Fisher's F probability distribution (continuous) (TI : RegM = d_1 = ; RegN = d_2 =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_F_page_1.pdf
	MENU	Generalised extreme value distribution		[GEV:]	GEV:	Generalised extreme value probability distribution (continuous) (TI : RegM = μ = ; RegS = σ = ; RegQ = ξ =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_GEV_page_1.pdf
	MENU	Logistic distribution		[Logis:]	Logis:	Logistic probability distribution (continuous) (TI : RegM = μ = ; RegS = s =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Logis_page_1.pdf
	MENU	Normal & Log normal distribution		[Norml:]	Norml:	Normal & Log normal probability distribution (continuous) (TI : RegM = μ = ; RegS = σ =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Norml_page_1.pdf
	MENU	Pareto distribution		[Pareto:]	Pareto:	Pareto probability distribution (continuous) -- see https://en.wikipedia.org/wiki/Pareto_distribution#Pareto_types_I-IV (TI : RegM = μ = ; RegS = σ = ; RegQ = α =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Pareto_page_1.pdf
	MENU	Standard normal distribution		[Φ :]	Φ :	Standard normal probability distribution (continuous) (Info : mean = 0 ; standard deviation = 1) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_StdNorml_page_1.pdf
	MENU	Student's t distribution		[t:]	t:	Student's t probability distribution (continuous) (TI : RegM = ν =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_t_page_1.pdf
	MENU	Uniform distribution		[Uniforms:]	Uniform:	Uniform probability distribution (continuous) (TI : RegM = a = ; RegN = b =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Uniform_page_1.pdf
	MENU	Weibull distribution		[Weibl:]	Weibl:	Weibull probability distribution (continuous) (TI : RegQ = k = ; RegS = λ =) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Weibl_page_1.pdf
	Function (monadic)	Cauchy cdf (lower)		Cauch _L	Cauch _L	Cauchy cumulative distribution (lower tail)
	Function (monadic)	Cauchy cdf (upper)		Cauch _U	Cauch _U	Cauchy cumulative distribution (upper tail)
	Function (monadic)	Cauchy pdf		Cauch _P	Cauch _P	Cauchy probability density function
	Function (monadic)	Cauchy qf (inverse)		Cauch ⁻¹	Cauch ⁻¹	Cauchy inverse cumulative distribution (quantile)
	Function (monadic)	Chi-squared cdf (lower) chi2		$\chi^2_L(x)$	$\chi^2_L(x)$	Chi-squared cumulative distribution (lower tail)
	Function (monadic)	Chi-squared cdf (upper) chi2		$\chi^2_U(x)$	$\chi^2_U(x)$	Chi-squared cumulative distribution (upper tail)
	Function (monadic)	Chi-squared pdf chi2		$\chi^2_P(x)$	$\chi^2_P(x)$	Chi-squared probability density function
	Function (monadic)	Chi-squared qf (inverse)		$(\chi^2)^{-1}$	$(\chi^2)^{-1}$	Chi-squared inverse cumulative distribution (quantile)
	Function (monadic)	Exponential cdf (lower)		Expon _L	Expon _L	Exponential cumulative distribution (lower tail)
	Function (monadic)	Exponential cdf (upper)		Expon _U	Expon _U	Exponential cumulative distribution (upper tail)
	Function (monadic)	Exponential pdf		Expon _P	Expon _P	Exponential probability density function
	Function (monadic)	Exponential qf (inverse)		Expon ⁻¹	Expon ⁻¹	Exponential inverse cumulative distribution (quantile)
	Function (monadic)	Fisher's F cdf (lower)		F _L (x)	F _L (x)	Fisher's F cumulative distribution (lower tail)
	Function (monadic)	Fisher's F cdf (upper)		F _U (x)	F _U (x)	Fisher's F cumulative distribution (upper tail)
	Function (monadic)	Fisher's F pdf		F _P (x)	F _P (x)	Fisher's F probability density function
	Function (monadic)	Fisher's F qf (inverse)		F ⁻¹ (p)	F ⁻¹ (p)	Fisher's F inverse cumulative distribution (quantile)
	Function (monadic)	Generalised extreme value cdf (lower)		GEV _L	GEV _L	Generalised extreme value cumulative distribution (lower tail)
	Function (monadic)	Generalised extreme value cdf (upper)		GEV _U	GEV _U	Generalised extreme value cumulative distribution (upper tail)
	Function (monadic)	Generalised extreme value pdf		GEV _P	GEV _P	Generalised extreme value probability density function
	Function (monadic)	Generalised extreme value qf (inverse)		GEV ⁻¹	GEV ⁻¹	Generalised extreme value inverse cumulative distribution (quantile)
	Function (monadic)	Log normal cdf (lower)		LgNrm _L	LgNrm _L	Log normal cumulative distribution (lower tail)
	Function (monadic)	Log normal cdf (upper)		LgNrm _U	LgNrm _U	Log normal cumulative distribution (upper tail)
	Function (monadic)	Log normal pdf		LgNrm _P	LgNrm _P	Log normal probability density function
	Function (monadic)	Log normal qf (inverse)		LgNrm ⁻¹	LgNrm ⁻¹	Log normal inverse cumulative distribution (quantile)
	Function (monadic)	Logistic cdf (lower)		Logis _L	Logis _L	Logistic cumulative distribution (lower tail)
	Function (monadic)	Logistic cdf (upper)		Logis _U	Logis _U	Logistic cumulative distribution (upper tail)
	Function (monadic)	Logistic pdf		Logis _P	Logis _P	Logistic probability density function
	Function (monadic)	Logistic qf (inverse)		Logis ⁻¹	Logis ⁻¹	Logistic inverse cumulative distribution (quantile)
	Function (monadic)	Normal cdf (lower)		Norml _L	Norml _L	Normal cumulative distribution (lower tail)
	Function (monadic)	Normal cdf (upper)		Norml _U	Norml _U	Normal cumulative distribution (upper tail)
	Function (monadic)	Normal pdf		Norml _P	Norml _P	Normal probability density function
	Function (monadic)	Normal qf (inverse)		Norml ⁻¹	Norml ⁻¹	Normal inverse cumulative distribution (quantile)
	Function (monadic)	Pareto type I cdf (lower)		Prto _L	Prto _L	Pareto type I cumulative distribution (lower tail)
	Function (monadic)	Pareto type I cdf (upper)		Prto _U	Prto _U	Pareto type I cumulative distribution (upper tail)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic)	Pareto type I pdf		Prto_p	Prto_p	Pareto type I probability density function
	Function (monadic)	Pareto type I qf (inverse)		Prto^{-1}_p	Prto^{-1}_p	Pareto type I inverse cumulative distribution (quantile)
	Function (monadic)	Pareto type II cdf (lower)		Prto2_ϕ	Prto2_ϕ	Pareto type II cumulative distribution (lower tail)
	Function (monadic)	Pareto type II cdf (upper)		Prto2_ϕ	Prto2_ϕ	Pareto type II cumulative distribution (upper tail)
	Function (monadic)	Pareto type II pdf		Prto2_p	Prto2_p	Pareto type II probability density function
	Function (monadic)	Pareto type II qf (inverse)		Prto2^{-1}_p	Prto2^{-1}_p	Pareto type II inverse cumulative distribution (quantile)
	Function (monadic)	Standard normal cdf (lower)		Φ_ϕ	Φ_ϕ	Standard normal cumulative distribution (lower tail)
	Function (monadic)	Standard normal cdf (upper)		Φ_ϕ	Φ_ϕ	Standard normal cumulative distribution (upper tail)
	Function (monadic)	Standard normal pdf		Φ_p	Φ_p	Standard normal probability density function
	Function (monadic)	Standard normal qf (inverse)		Φ^{-1}	Φ^{-1}	Standard normal inverse cumulative distribution (quantile)
	Function (monadic)	Student's t cdf (lower)		$t_\phi(x)$	$t_\phi(x)$	Student's t cumulative distribution (lower tail)
	Function (monadic)	Student's t cdf (upper)		$t_\phi(x)$	$t_\phi(x)$	Student's t cumulative distribution (upper tail)
	Function (monadic)	Student's t pdf		$t_\phi(x)$	$t_\phi(x)$	Student's t probability density function
	Function (monadic)	Student's t qf (inverse)		$t^{-1}(p)$	$t^{-1}(p)$	Student's t inverse cumulative distribution (quantile)
	Function (monadic)	Uniform cdf (lower)		Unfm_ϕ	Unfm_ϕ	Uniform cumulative distribution (lower tail)
	Function (monadic)	Uniform cdf (upper)		Unfm_ϕ	Unfm_ϕ	Uniform cumulative distribution (upper tail)
	Function (monadic)	Uniform pdf		Unfm_p	Unfm_p	Uniform probability density function
	Function (monadic)	Uniform qf (inverse)		Unfm^{-1}	Unfm^{-1}	Uniform inverse cumulative distribution (quantile)
	Function (monadic)	Weibull cdf (lower)		Weibl_ϕ	Weibl_ϕ	Weibull cumulative distribution (lower tail)
	Function (monadic)	Weibull cdf (upper)		Weibl_ϕ	Weibl_ϕ	Weibull cumulative distribution (upper tail)
	Function (monadic)	Weibull pdf		Weibl_p	Weibl_p	Weibull probability density function
	Function (monadic)	Weibull qf (inverse)		Weibl^{-1}	Weibl^{-1}	Weibull inverse cumulative distribution (quantile)
	Command	Store d_1 in M		$\text{STO}_M d_1$		Store parameter d_1 : degree of freedom in register M (Split screen : $\text{RegM} = d_1 =$)
	Command	Store d_2 in N		$\text{STO}_N d_2$		Store parameter d_2 : degree of freedom in register N (Split screen : $\text{RegN} = d_2 =$)
	Command	Store k in Q		$\text{STO}_Q k$		Store parameter k : shape in register Q (Split screen : $\text{RegQ} = k =$)
	Command	Store s in S		$\text{STO}_S s$		Store parameter s : scale in register S (Split screen : $\text{RegS} = s =$)
	Command	Store x_0 in M		$\text{STO}_M x_0$		Store parameter x_0 : location in register M (Split screen : $\text{RegM} = x_0 =$)
	Command	Store α in Q		$\text{STO}_Q \alpha$		Store parameter α : shape in register Q (Split screen : $\text{RegQ} = \alpha =$)
	Command	Store γ in S		$\text{STO}_S \gamma$		Store parameter γ : scale in register S (Split screen : $\text{RegS} = \gamma =$)
	Command	Store λ in R		$\text{STO}_R \lambda$		Store parameter λ : rate parameter in register R (Split screen : $\text{RegR} = \lambda =$)
	Command	Store λ in S		$\text{STO}_S \lambda$		Store parameter λ : scale in register S (Split screen : $\text{RegS} = \lambda =$)
	Command	Store μ in M		$\text{STO}_M \mu$		Store parameter μ : location in register M (Split screen : $\text{RegM} = \mu =$)
	Command	Store ν in M		$\text{STO}_M \nu$		Store parameter ν : degrees of freedom in register M (Split screen : $\text{RegM} = \nu =$)
	Command	Store ξ in Q		$\text{STO}_Q \xi$		Store parameter ξ : shape in register Q (Split screen : $\text{RegQ} = \xi =$)
	Command	Store σ in S		$\text{STO}_S \sigma$		Store parameter σ : scale in register S (Split screen : $\text{RegS} = \sigma =$)
Programming	Variable (longint)	Graphics display mode	0	GRAMOD	GRAMOD	Reserved LongInteger variable for the graphics display mode (GRAMOD) ; set interactively from menu P.FN1 (AGRAPH) (Info : OR: 0 ; SET: 1 ; OFF: 2 ; XOR: 3)
	MENU	Flags		[FLAG]	FLAG	Setting, clearing and testing flags ; Keyboard f[6] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FLAG_page_1.pdf
	MENU	Looping		[LOOP]	LOOP	Looping (programming) functions ; Keyboard g[3] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_LOOP_page_1.pdf
	MENU	Programming functions		[P.FN]	P.FN	User menu to quickly access selected menus and functions for programming ; all buttons are user assignable ('riveted' look) ; initially populated for basic programming options ; reset using PFN.R (Shortcut : PEM long [EXIT] ; PEM triple [f/g] (because either HOME ϕ or MyM ϕ is SET and either HOME ϕ or MyMb is SET) ; PEM longer [f/g] (because either HOME ϕ or MyM ϕ is SET and either HOME ϕ or MyMb is SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_P-FN_page_1.pdf
	MENU	Programming functions 1		[P.FN1]	P.FN1	Programming functions 1 ; this is a normal menu, buttons are not user assignable (P.FN is) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_P-FN1_page_1.pdf
	MENU	Programming functions 2		[P.FN2]	P.FN2	Programming functions 2 ; this is a normal menu, buttons are not user assignable (P.FN is) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_P-FN2_page_1.pdf
	MENU	Programming functions 3		[P.FN3]	P.FN3	Programming functions 3 ; this is a normal menu, buttons are not user assignable (P.FN is) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_P-FN3_page_1.pdf
	MENU	Testing		[TEST]	TEST	Testing functions ; Keyboard g[R/S] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TEST_page_1.pdf
	Flag (system)	Variable menu displayed	CLEAR	VMDISP		SET for variable menu displayed ; CLEAR for variable menu not displayed
	Flag	Auto execute program	CLEAR	AUTXEQ		SET for auto execute program ; CLEAR for auto execute program inactive (Info : Automatically starts program when calculator turns on)
	Flag	Ignore one error	CLEAR	IGNIER		SET for calculator ignoring next error ; CLEAR for calculator not ignoring error(s) (Info : IGNIER is cleared subsequently)
	Command (PEM)	Allocate local registers		LocR	LocR	Allocate nn (0-99) local registers and 32 local flags (0-31) ; available in the current routine (TAM : LocR __ TamNonReg menu ; Info : LocR nn (nn > 0) allocates local registers 00..nn-1 and local flags 00..31, LocR 00 allocates only the local flags)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (PEM)	Alpha graphics		AGRAPH	AGRAPH	Show a 64 column pattern according to variable GRAMOD (OR: 0 ; SET: 1 ; OFF: 2 ; XOR: 3) ; this command will be redesigned and should not be used in programming! (TAM : AGRAPH __ Tam menu)
	Command (PEM)	Alpha view		AVIEW	AVIEW	View the register or variable contents (text string) in the top left part of the screen, without TI (for use in Program Entry Mode) ; if the contents include the ¶ symbol, optionally display one or more TI's at the top, X, Y, Z line ; example: "top line = ¶ X = ¶ Y = ¶ Z = " (TAM : AVIEW __ Tam menu ; Info : if ¶ACT is SET, AVIEW will print when executed from the keyboard in MAN, NORM or TRACE ; in Program Entry Mode, ¶EN must be SET as well ; if ¶ACT is CLEAR and ¶EN is SET, AVIEW halts execution (R/S))
	Command (PEM)	Back		BACK	BACK	Jumps back n steps in program (TAM : BACK __ TamNonReg menu)
	Command (PEM)	Case		CASE	CASE	Case statement, skips number of steps given by chosen variable (normally followed by number of GTO statements) (TAM : CASE __ Tam menu)
	Command (PEM)	Clear registers		R-CLR	R-CLR	Clear registers, using range number in X (sss.nn means registers from sss through sss + nn - 1)
	Command (PEM)	Copy registers		R-COPY	R-COPY	Copy registers, using range number in X (sss.nndd means registers from sss through sss + n - 1 to registers ddd through ddd + nn - 1)
	Command (PEM)	Define menu variable		MVAR	MVAR	Define menu variable in RPN program, for VarMNU, integrator, or solver ; creates variable if it does not exist (TAM : MVAR __ TamVarOnly menu ; Ref : https://47calc.com/doc/C47/Reference/C47_Variable_naming.pdf)
	Command (PEM)	Display pixel or thin line		PIXEL	PIXEL	Display one pixel at (X, Y) or draw a thin horizontal or vertical line at the (absolute) value of one or both negative coordinates ; this is a Program Entry Mode command ; use CLLCD to clear LCD and PAUSE to pause program for number of ticks in order to view graphic (Info : Range is X in [0, 399] and Y in [0, 239] (400 x 240 display, where (0, 0) is bottom left) ; -0. is accepted as a valid negative input for line drawing)
	Command (PEM)	Display point or thick line		POINT	POINT	Display 9 (3x3) pixels around (X, Y) or draw a thick horizontal or vertical line at the (absolute) value of one or both negative coordinates ; this is a Program Entry Mode command ; use CLLCD to clear LCD and PAUSE to pause program for number of ticks in order to view graphic (Info : Range is X in [0, 399] and Y in [0, 239] (400 x 240 display, where (0, 0) is bottom left) ; -0. is accepted as a valid negative input for line drawing)
	Command (PEM)	Display Programmable MENU		MENU	MENU	Display the Programmable MENU defined with KEYG and/or KEYX
	Command (PEM)	End		END	END	End statement
	Command (PEM)	Error		ERR	ERR	Raise error and show error message (TAM : ERR __ TamNonReg menu ; Ref : https://47calc.com/doc/C47/Reference/C47_Errors_and_messages.pdf)
	Command (PEM)	Execute		XEQ	XEQ	Execute function or program with global or local label (normal spaces can be used instead of special spaces in labels) (TAM : XEQ __ TamLabel menu ; Info : XEQ + TAM [XEQ] to enter global XEQ '_. TamAlpha menu ; XEQ + button : on [TamLabel.1]F3 to enter XEQ :__ TamLocalLabel menu for easy access to local labels a-l, D-F or local named labels (additionally pressing button α on [TamLocalLabel.1]F3 or TAM [XEQ] to enter XEQ :_. TamAlpha menu)) ; Keyboard [XEQ]
	Command (PEM)	Execute and skip		XEQ.SKP	XEQ.SKP	Execute function or program with global or local label (normal spaces can be used instead of special spaces in labels) and skip the next instruction (intended use is after a test function, which if true, calls XEQ.SKP and jumps over the next program step, which means that step is executed only if the test function returns false) (Ref : https://47calc.com/doc/C47/Reference/C47_Skipping_program_steps.pdf)
	Command (PEM)	Exit all		EXITall	EXITALL	Exit all menus ; return to MyMenu (Normal Mode) ; exit VarMNU (program execution)
	Command (PEM)	Go to		GTO	GTO	Go to global or local label ; GTO. to go to (local) label or step ; GTO.. to move the program pointer to the end of program memory ; GTO↑ to go to the start of the (previous) program or jump to the start of program memory ; GTO↓ to go to the next program or jump to the end of program memory (TAM : GTO __ TamLabel menu ; Info : GTO + TAM [XEQ] to enter global GTO '_. TamAlpha menu ; GTO + button : on [TamLabel.1]F3 to enter GTO :__ TamLocalLabel menu for easy access to local labels a-l, D-F or local named labels (additionally pressing button α on [TamLocalLabel.1]F3 or TAM [XEQ] to enter GTO :_. TamAlpha menu)) ; Keyboard g[XEQ]
	Command (PEM)	Input		INPUT	INPUT	Halt program execution, push current value, prompt and accept input for register or variable (TAM : INPUT __ Tam menu)
	Command (PEM)	Key		KEY		Used in Program Entry Mode to specify soft button (1-18) of the Programmable MENU, forming program step for KEYG → KEY nn GTO ; KEYX → KEY nn XEQ (TAM : KEY __ TamKey menu)
	Command (PEM)	Key Execute		KEYX	KEYX	Used in Program Entry Mode to specify the program to execute when soft button (1-18) of the Programmable MENU, [▲] (19), [▼] (20) or [EXIT] (21) is pressed ; if button 1-18 is specified, KEYX also creates a soft button in the Programmable MENU with the string in X as a label (TAM : KEYX → KEY __ TamKey menu ; KEY nn XEQ __ TamLabel menu)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (PEM)	Key Go to		KEYG	KEYG	Used in Program Entry Mode to specify the label to go to when soft button (1-18) of the Programmable MENU, [▲] (19), [▼] (20) or [EXIT] (21) is pressed ; if button 1-18 is specified, KEYG also creates a soft button in the Programmable MENU with the string in X as a label (TAM : KEYG → KEY ___ TamKey menu ; KEY nn GTO ___ TamLabel menu)
	Command (PEM)	Label		LBL	LBL	Create global or local label ; create global label by pressing key α (f[XEQ]) and entering the name, which will be displayed as '<label>' ; local labels only exist within the scope of the program ; they are numeric : 01-99, single letter : a-l or {A}-{E}, F, {G}-{L} ; or created as local named labels, by pressing button : ([TamLabel.1]F3), then button α ([TamLocalLabel.1]F3) or key α (f[XEQ]) and entering the name, which will be displayed as '<label>' ; maximum label length is : 7 characters (TAM : LBL ___ TamLabel menu ; Info : LBL + TAM [XEQ] to enter global LBL '___ TamAlpha menu ; LBL + button : on [TamLabel.1]F3 to enter LBL :___ TamLocalLabel menu to create local labels a-l, D-F or local named labels (additionally pressing button α on [TamLocalLabel.1]F3 or TAM [XEQ] to enter LBL :___ TamAlpha menu) ; note that a label may be used multiple times in a program ; local labels are searched forwards from the GTO or XEQ command, before global labels, which are searched from the start of program memory) ; Keyboard g[LN]
	Command (PEM)	Message		MSG	MSG	Show error message (TAM : MSG ___ TamNonReg menu ; Ref : https://47calc.com/doc/C47/Reference/C47_Errors_and_messages.pdf)
	Command (PEM)	No operation		NOP	NOP	No operation (empty step) ; Keyboard longestpress [Σ+]
	Command (PEM)	Open menu		OPENM	OPENM	Open menu by (catalog) name, optionally followed by ↵ and page number (TAM : OPENM _ TamMenu menu ; Info : ↵ is on [Misc.1]F1 ; menu name and page can be stored in a register or variable ; Error 59 -- Undefined menu name -- may be raised)
	Command (PEM)	Pause		PAUSE	PAUSE	Pause program for number of ticks ; range is nn in [0, 99], one tick is 100 ms ; special case : nn = 99 results in sleep mode (TAM : PAUSE ___ TamNonReg menu)
	Command (PEM)	Pop local registers		PopLR	PopLR	Pop local registers (no return to calling routine)
	Command (PEM)	Prompt		PROMPT	PROMPT	Halt program execution and view the register or variable contents (text string) in the top left part of the screen, without TI (for use in Program Entry Mode) ; if the contents include the ↵ symbol, optionally display one or more TI's at the top, X, Y, Z line ; example: "top line = ↵ X = ↵ Y = ↵ Z = " (TAM : PROMPT ___ Tam menu ; Info : if ⌨EN is SET, PROMPT will print when executed from the keyboard in NORM or TRACE ; in Program Entry Mode, ⌨EN must be SET as well)
	Command (PEM)	Put keycode in buffer		PUTK	PUTK	Copy keycode from register to keyboard buffer for immediate execution (TAM : PUTK ___ Tam menu)
	Command (PEM)	Recall and skip		RCL.SKP	RCL.SKP	Recall value from register or variable and skip the next instruction (intended use is after a test function, which if true, calls RCL.SKP and jumps over the next program step, which means that step is executed only if the test function returns false) (Ref : https://47calc.com/doc/C47/Reference/C47_Skipping_program_steps.pdf)
	Command (PEM)	Remark		REM	REM	Comment line in program (maximum 160 characters) (Info : REM '___ ALPHA (ALPHA) menu)
	Command (PEM)	Return		RTN	RTN	Return from subroutine to calling program
	Command (PEM)	Return and skip		RTN.SKP	RTN.SKP	Return from subroutine to calling program and skip the instruction following the calling XEQ (intended use is to allow a subroutine to function as a built in test function, which if true, returns using RTN and using RTN.SKP otherwise) (Ref : https://47calc.com/doc/C47/Reference/C47_Skipping_program_steps.pdf)
	Command (PEM)	Skip		SKIP	SKIP	Skip n program steps (TAM : SKIP ___ TamNonReg menu ; Ref : https://47calc.com/doc/C47/Reference/C47_Skipping_program_steps.pdf)
	Command (PEM)	Sort registers		R-SORT	R-SORT	Sort registers by ascending value, using range number in X (sss.nn means registers from sss through sss + nn - 1) (Info : R-SORT can also sort by time value stored in the registers)
	Command (PEM)	Swap registers		R-SWAP	R-SWAP	Swap registers, using range number in X (sss.nndd means registers from sss through sss + n - 1 to registers ddd through ddd + nn - 1)
	Command (PEM)	Ticks		TICKS	TICKS	Number of ticks counted since calculator was turned on (one tick is 100 ms)
	Command (PEM)	Variable menu		VarMNU	VARMNU	Create variable menu ; uses register K to store variable name when MVAR-defined button (<var...>) is pressed without NIM input (TAM : VarMNU ___ TamLblOnly menu ; Error 25 -- No MVAR found in selected program -- may be raised)
	Command (PEM)	Zoom (plot)		ZOOM	ZOOM	Zoom plot to factor in X (Info : -3..-1 : zoom out ; 0 : normal)
	Command (PEM, old)	Message		>MSG<		Old version of the command MSG that used indirect addressing (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use MSG on [P.FN1.1]fF1 instead)
	Command (nonpgm)	Backstep		BST	BST	Backstep (Shortcut : [▲])
	Command (nonpgm)	Single step		SST	SST	Single step (Shortcut : [▼])
	Command (logical)	(Data) entry?		ENTRY?	ENTRY?	Test internal entry flag for data entry

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (logical)	2D vector?		VECT2D?	VECT2D?	Test X is a 2D vector (test function MATR? is true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	3D vector?		VECT3D?	VECT3D?	Test X is a 3D vector (test function MATR? is true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Angle?		ANGLE?	ANGLE?	Test type of X is Angle (1.1-1.5) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Complex matrix or vector?		Cmatx?	Cmatx?	Test type of X is Complex matrix or vector (7.n) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Complex number?		C?	C?	Test type of X is Complex number (2.0-2.5) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Config?		Config?	Config?	Test type of X is Config (9) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Convergence?		CONVG?	CONVG?	Test convergence of X and Y using binary coded parameter for comparison mode and special numbers ; tolerance is derived from setting SDIGS (TAM : CONVG? _ TamNonReg menu ; Ref : https://47calc.com/doc/C47/Reference/C47_CONVG_param.pdf)
	Command (logical)	Date?		D?	D?	Test type of X is Date (4) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Even?		EVEN?	EVEN?	Test X is even and integer (test function INT? is true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Flag clear?		FC?	FC?	Test flag clear? (TAM : FC? _ TamFlag menu)
	Command (logical)	Flag clear? and clear		FC?C	FC?C	Test flag clear? and clear (TAM : FC?C _ TamFlag menu)
	Command (logical)	Flag clear? and flip		FC?F	FC?F	Test flag clear? and flip (TAM : FC?F _ TamFlag menu)
	Command (logical)	Flag clear? and set		FC?S	FC?S	Test flag clear? and set (TAM : FC?S _ TamFlag menu)
	Command (logical)	Flag set?		FS?	FS?	Test flag set? (TAM : FS? _ TamFlag menu)
	Command (logical)	Flag set? and clear		FS?C	FS?C	Test flag set? and clear (TAM : FS?C _ TamFlag menu)
	Command (logical)	Flag set? and flip		FS?F	FS?F	Test flag set? and flip (TAM : FS?F _ TamFlag menu)
	Command (logical)	Flag set? and set		FS?S	FS?S	Test flag set? and set (TAM : FS?S _ TamFlag menu)
	Command (logical)	Fractional Part?		FP?	FP?	Test X has nonzero fractional part and is Real number (1.0-1.5) (test function R? is true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Infinities?		±∞?	±∞?	Test X contains positive or negative infinite value
	Command (logical)	Integer?		INT?	INT?	Test X is short integer (8.02-8.16), long integer (0) or Real number (1.0-1.5) with zero fractional part (test function SINT?, LINT? or R? is true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Key type?		KTYP	KTYP	Get key type for key code returned by KEY? (TAM : KTYP _ Tam menu ; Info : https://47calc.com/doc/C47/Reference/C47_KeyType.pdf)
	Command (logical)	Label exists?		LBL?	LBL?	Test label exists (TAM : LBL? _ TamLabel menu)
	Command (logical)	Leap year?		LEAP?	LEAP?	Test X is a date in a leap year (test function D? must be true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Long integer?		LINT?	LINT?	Test type of X is Long integer (0) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Non-zero imaginary part?		Im≠0?	Im≠0?	Test X has imaginary part not equal to zero
	Command (logical)	Non-zero real part?		Re≠0?	Re≠0?	Test X has real part not equal to zero
	Command (logical)	not-a-number?		NaN?	NaN?	Test X is not-a-number
	Command (logical)	Number?		NUMBR?	NUMBR?	Test type of X is Number (0 ; 1.0-1.5 ; 2.0-2.5) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Odd?		ODD?	ODD?	Test X is odd and integer (test function INT? is true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Primality test		PRIME?	PRIME?	Test X is a prime number (must be integer) -- see https://en.wikipedia.org/wiki/Miller%E2%80%93Rabin_primality_test (Info : Miller-Rabin probabilistic primality test is used (25 repetitions) ; 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))
	Command (logical)	Real matrix or vector?		MATR?	MATR?	Test type of X is Real matrix or vector (6) or Complex matrix or vector (7.n) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Real matrix or vector?		Rmatx?	Rmatx?	Test type of X is Real matrix or vector (6) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Real number?		R?	R?	Test type of X is Real number (1.0-1.5) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Short integer?		SINT?	SINT?	Test type of X is Short integer (8.02-8.16) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Special?		SPEC?	SPEC?	Test X is special (±∞ or NaN)
	Command (logical)	Square matrix?		M.SQR?	M.SQR?	Test X is a square matrix (test function MATR? is true) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Text string?		STRI?	STRI?	Test type of X is Text string (5) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)
	Command (logical)	Time?		T?	T?	Test type of X is Time (3) (Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (logical)	Top?		TOP?	TOP?	Test program pointer is in top routine (as opposed to in subroutine)
	Command (logical)	X approximates?		$x \approx ?$	$x \approx ?$	Test X approximates register or variable contents (rounded values are equal) (TAM : $x \approx ?$ __ TamCmp menu)
	Command (logical)	X equals -0?		$x = -0 ?$	$x = -0 ?$	Test X equals -0 (Real number)
	Command (logical)	X equals +0?		$x = +0 ?$	$x = +0 ?$	Test X equals +0 (Real number)
	Command (logical)	X equals?		$x = ?$	$x = ?$	Test X equals register or variable contents (TAM : $x = ?$ __ TamCmp menu)
	Command (logical)	X greater than or equal to +0?		$x \geq +0 ?$	$x \geq +0 ?$	Test X is greater than or equal to +0 (Real number)
	Command (logical)	X greater than or equal to?		$x \geq ?$	$x \geq ?$	Test X is greater than or equal to register or variable contents (TAM : $x \geq ?$ __ TamCmp menu)
	Command (logical)	X greater than?		$x > ?$	$x > ?$	Test X is greater than register or variable contents (TAM : $x > ?$ __ TamCmp menu)
	Command (logical)	X less than or equal to -0?		$x \leq -0 ?$	$x \leq -0 ?$	Test X is less than or equal to -0 (Real number)
	Command (logical)	X less than or equal to?		$x \leq ?$	$x \leq ?$	Test X is less than or equal to register or variable contents (TAM : $x \leq ?$ __ TamCmp menu)
	Command (logical)	X less than?		$x < ?$	$x < ?$	Test X is less than register or variable contents (TAM : $x < ?$ __ TamCmp menu)
	Command (logical)	X not equal to?		$x \neq ?$	$x \neq ?$	Test X is not equal to register or variable contents (TAM : $x \neq ?$ __ TamCmp menu)
	Command (logical)	Zero imaginary part?		Im=0?	Im=0?	Test X has imaginary part equal to zero
	Command (logical)	Zero real part?		Re=0?	Re=0?	Test X has real part equal to zero
	Command (logical, PEM)	Key pressed?		KEY?	KEY?	Test key was pressed : if it is NOT pressed, the next program step will be executed and if it IS pressed, the next program step will be skipped and the keycode is stored in the register specified ; keycode range is 11-85, where the first digit is row, second digit is column number (TAM : KEY? __ Tam menu ; Info : KEY? should not to be used in short loops, keeps calculator running on full power ; [R/S] and [EXIT] interrupt a program and KEY? will not work for those keys ; Ref : https://47calc.com/doc/C47/Reference/C47_KeyType.pdf)
	Command	Clear flag		CF	CF	Clear flag (TAM : CF __ TamFlag menu)
	Command	Decrement		DECR	DECR	Decrement range number in register or variable (ccc.ffffd, where ccc is counter, fff is final value, dd is decrement), by decrement (TAM : DECR __ Tam menu)
	Command	Decrement skip on equal		DSE	DSE	Decrement skip on equal, using range number in register or variable (ccc.ffffd, where ccc is counter, fff is final value, dd is decrement) (TAM : DSE __ Tam menu)
	Command	Decrement skip on less		DSL	DSL	Decrement skip on less, using range number in register or variable (ccc.ffffd, where ccc is counter, fff is final value, dd is decrement) (TAM : DSL __ Tam menu)
	Command	Decrement skip on zero		DSZ	DSZ	Decrement skip on zero, using range number in register or variable (ccc.ffffd, where ccc is counter, fff is final value, dd is decrement) (TAM : DSZ __ Tam menu ; Info : time values may be used as the range number, decrement is one hour)
	Command	Flip flag		FF	FF	Flip flag (toggle) (TAM : FF __ TamFlag menu)
	Command	Get current menu		MENU#	MENU#	Get current menu name ; followed by # and page number if 2 or higher
	Command	Increment		INCR	INCR	Increment range number in register or variable (ccc.ffffi, where ccc is counter, fff is final value, ii is increment), by increment (TAM : INCR __ Tam menu)
	Command	Increment skip on equal		ISE	ISE	Increment skip on equal, using range number in register or variable (ccc.ffffi, where ccc is counter, fff is final value, ii is increment) (TAM : ISE __ Tam menu)
	Command	Increment skip on greater		ISG	ISG	Increment skip on greater, using range number in register or variable (ccc.ffffi, where ccc is counter, fff is final value, ii is increment) (TAM : ISG __ Tam menu)
	Command	Increment skip on zero		ISZ	ISZ	Increment skip on zero, using range number in register or variable (ccc.ffffi, where ccc is counter, fff is final value, ii is increment) (TAM : ISZ __ Tam menu ; Info : time values may be used as the range number, increment is one hour)
	Command	Programming		PRGM		Enter Program Entry Mode (PEM) and activate menu P.FN (Info : Program Entry Mode starts in UPPERcase) ; Keyboard f[R/S]
	Command	Run/Stop program		R/S		Run/Stop program (Info : In Program Entry Mode, R/S enters STOP) ; Keyboard [R/S]
	Command	Set flag		SF	SF	Set flag (TAM : SF __ TamFlag menu)
	Command	Stop program		R/S	STOP	Stop program (Info : In Program Entry Mode, R/S enters STOP)
	Button (black)	Button 1		1		Soft button 1 for selection by KEYG or KEYX command
	Button (black)	Button 10		10		Soft button 10 for selection by KEYG or KEYX command
	Button (black)	Button 11		11		Soft button 11 for selection by KEYG or KEYX command
	Button (black)	Button 12		12		Soft button 12 for selection by KEYG or KEYX command
	Button (black)	Button 13		13		Soft button 13 for selection by KEYG or KEYX command
	Button (black)	Button 14		14		Soft button 14 for selection by KEYG or KEYX command
	Button (black)	Button 15		15		Soft button 15 for selection by KEYG or KEYX command
	Button (black)	Button 16		16		Soft button 16 for selection by KEYG or KEYX command
	Button (black)	Button 17		17		Soft button 17 for selection by KEYG or KEYX command
	Button (black)	Button 18		18		Soft button 18 for selection by KEYG or KEYX command
	Button (black)	Button 2		2		Soft button 2 for selection by KEYG or KEYX command
	Button (black)	Button 3		3		Soft button 3 for selection by KEYG or KEYX command
	Button (black)	Button 4		4		Soft button 4 for selection by KEYG or KEYX command
	Button (black)	Button 5		5		Soft button 5 for selection by KEYG or KEYX command
	Button (black)	Button 6		6		Soft button 6 for selection by KEYG or KEYX command
	Button (black)	Button 7		7		Soft button 7 for selection by KEYG or KEYX command

Category	Type	FullName	Default	Label	Catalog	Extended description
	Button (black)	Button 8		8		Soft button 8 for selection by KEYG or KEYX command
	Button (black)	Button 9		9		Soft button 9 for selection by KEYG or KEYX command
Reals	Setting (pgm)	Set number range	6145	RNG	RNG ₆₁₄₅	Set number range as 10^{nn} , where nn in [99, 6145] (TAM : RNG ____ TamNonReg menu ; Info : numbers smaller than 10^{-RNG} (or the value set by HIDE) will be displayed as ≈ 0 ; numbers greater than 10^{RNG} will be displayed as ' ∞ ' (or ∞ if RNG is 6145 and provided flag SPGRES is SET))
	Setting (pgm)	Set rounding mode	0	RMODE	RMODE ₀	Set floating point rounding mode ; used for RSD and RDP ; also used when converting from the extended precision internal format to packed reals ; 0 : $\frac{1}{2}E$ = round half even ; 1 : $\frac{1}{2}\uparrow$ = round half up ; 2 : $\frac{1}{2}\downarrow$ = round half down ; 3 : $\leftarrow 0$ = round away from zero ; 4 : $\rightarrow 0$ = round towards zero ; 5 : $\lceil x \rceil$ = ceiling ; 6 : $\lfloor x \rfloor$ = floor (TAM : RMODE _ TamNonReg menu ; Info : may be abbreviated as RM in status overviews)
	Setting (pgm, old)	Set number range	6145	>RNG<		Old version of the setting RNG that could not use indirect addressing (Info : May be found in older user programs, use RNG on [DISP.2] gF5 instead)
	MENU	Real		[REAL]	REAL	Functions on Real numbers and Complex numbers ; Keyboard g[6] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_REAL_page_1.pdf
	Function (monadic)	Ceiling		ceil	ceil	Mathematical ceiling of number
	Function (monadic)	Convert to Real number		.d	$\rightarrow$ REAL	Convert to Real number (1.0-1.5)
	Function (monadic)	Decimal (Real) or Date		.d		Convert to decimal (Real) or date value ; untag (ADM) angle ; clear Fractions Mode, #BASE Mode ; convert angles / Times / Dates to Real ; convert NIM input to Date (according to CLK date format settings and implied conversion set by YY ; convert Complex number with zero imaginary part to Real ; when editing angles using EDIT, key .d removes the angle tag (Normal Mode and PEM) ; when editing short integers, key .d removes the base (Normal Mode and PEM) ; in PEM, .d registers as $\rightarrow$ REAL (TI : decimal h: or yyyy.mmmdd ; Ref : https://47calc.com/doc/C47/Reference/C47_Date_formats.pdf) ; Keyboard g[L06]
	Function (monadic)	Decompose		DECOMP	DECOMP	Convert (improper) real number fraction in X to nominator in Y, denominator in X (Info : Honours settings DENANY, DENFIX and DMX)
	Function (monadic)	Exponent		EXPT	EXPT	Exponent of real number
	Function (monadic)	Floor		floor	floor	Mathematical floor of number
	Function (monadic)	Mantissa		MANT	MANT	Mantissa of real number
	Function (monadic)	Round		ROUND	ROUND	Round number to current display format (returns Real number)
	Function (monadic)	Round to decimal places		RDP	RDP	Round real number to number of decimal places, subject to rounding mode (RMODE) (TAM : RDP __ TamNonReg menu ; Range is [0, 34])
	Function (monadic)	Round to integer		ROUNDI	ROUNDI	Round real number to nearest long integer (max 1000 digits)
	Function (monadic)	Round to significant digits		RSD	RSD	Round real number to number of significant digits, subject to rounding mode (RMODE) (TAM : RSD __ TamNonReg menu)
	Function (monadic)	Shift Digits Left		SDL	SDL	Shift digits of Integer Part of real number to the left (zeros are added) (TAM : SDL __ TamNonReg menu)
	Function (monadic)	Shift Digits Right		SDR	SDR	Shift digits of Integer Part of real number to the right (digits are dropped) (TAM : SDR __ TamNonReg menu)
	Function (monadic)	Sign		sign	sign	Sign of number (Info : Sign is -1 for negative numbers, 0 for zero, +1 for positive numbers ; for nonzero Complex numbers, $\text{sign}(z) = z / z $ (a.k.a. UNITV))
	Function (monadic, old)	Convert to decimal hours		$\rightarrow$ HR<		Old function $\rightarrow$ HR that assumed argument conforms to ADM (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use .d on g[L06] or [CLK.1] fF5 instead)
	Command	Get number range		RNG#	RNG#	Get number range as 10^{nn} , where nn in [99, 6145] (Info : Set by RNG)
	Command	Get real number display format	0	DF#	DF#	Get real number display format ; set interactively from menu MLoads: (formats FIX ; SCI ; ENG ; UNIT ; SIG ; ALL) and by DF (Info : FIX : 1 ; SCI : 2 ; ENG : 3 ; UNIT : 5 ; SIG : 4 ; ALL : 0)
	Command	Get rounding mode	0	RMODE#	RMODE#	Get floating point rounding mode (Info : Set by RMODE)
	Command	Set real number display format	0	DF	DF ₀	Set real number display format to X ; also set interactively from menu MLoads: (formats FIX ; SCI ; ENG ; UNIT ; SIG ; ALL) (Info : FIX : 1 ; SCI : 2 ; ENG : 3 ; UNIT : 5 ; SIG : 4 ; ALL : 0)
Registers	Command	Get number of local registers		LocR#	LocR#	Get number of local registers (current routine) (Info : Set by LocR)
RPN Calculus and plotting	MENU (item)	Plot (RPN)		[PLT f]	PLT f	Plot the RPN program (with MVAR variables) selected with TamLbOnly menu ; presents menu for variable selection ; in Program Entry Mode, the RPN program must be specified by PGMPLOT and the variable to plot for using the TamVarOnly menu (TAM : PLT f _ TamLbOnly menu (uses TamVarOnly menu in PEM, to specify variable to plot for) ; Info : in Program Entry Mode, use PAUSE, SNAP or PLTFCNS to view the plot ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PLTf_page_1.pdf
	MENU	Advanced		[ADV]	ADV	Advanced functions, using RPN user programs (Ref : https://47calc.com/doc/Application%20Notes/AN0011_C47_JM_EQN_and_ADV_menu_solver_integrator_updated.pdf) ; Keyboard f[8] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ADV_page_1.pdf

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU	Integrator (RPN)		$\int f d$	$\int f d$	Integrator (variables) menu for the RPN program (with MVAR variables) selected with TamLbLOnly menu ; enter value and press <var> button(s) to set variables, then press button for variable to be used by the integrator in menu Tool ; in Program Entry Mode, the RPN program must be specified by PGMINIT and the variable with the Tam menu ; tolerance set by variable ACC (TAM : $\int f d$ _ TamLbLOnly menu (uses Tam menu in PEM, to specify variable to integrate for)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Integral-fd_page_1.pdf
	Command (submnu)	First derivative (RPN)		f''	f''	Numerically calculate first derivative using RPN program, at value of starred MVAR variable and using step size of δ_4 , as specified using the buttons in the (MVAR (delta)) menu presented (TAM : f'' _ TamLbLOnly menu ; TI : $\delta =$; $f'' \approx$; Info : if no step size is entered, an automatic step size is determined by iteration in 16 steps, starting with 1/10 ; the step size will always be displayed as " δ ")
	Command (submnu)	Second derivative (RPN)		f'	f'	Numerically calculate second derivative using RPN program, at value of starred MVAR variable and using step size of δ_4 , as specified using the buttons in the (MVAR (delta)) menu presented (TAM : f' _ TamLbLOnly menu ; TI : $\delta =$; $f' \approx$; Info : if no step size is entered, an automatic step size is determined by iteration in 16 steps, starting with 1/10 ; the step size will always be displayed as " δ ")
	Command (submnu)	Solver (RPN)		SOLVE	SOLVE	Real solver for the RPN program (with MVAR variables) selected with TamLbLOnly menu ; presents menu for initial guesses and variable selection ; interrupt by [EXIT] ; tolerance set by SDIGS ; in Program Entry Mode, the RPN program must be specified by PGMSLV and the variable to solve for using the TamVarOnly menu ; SOLVE will skip the next program step if no root is found (TAM : SOLVE _ TamLbLOnly menu (uses TamVarOnly menu in PEM, to specify variable to solve for) ; Info : displays current top and bottom results and iteration counter ; shown if flag MONIT is SET ; Displays : Accuracy $\approx$; <var1..n> $\approx$)
	Command (old)	First derivative (RPN)		$>f'(x)<$		Old version of the command to numerically calculate the first derivative at X using RPN program (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use f' on [ADV.1]F6 instead)
	Command (old)	Second derivative (RPN)		$>f''(x)<$		Old version of the command to numerically calculate the second derivative at X using RPN program (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use f'' on [ADV.1]gF6 instead)
	Command	Infinite sum (RPN)		$\omega \sum_n$	$\omega \sum_n$	Approximate infinite sum using RPN program ; $\omega \sum_n$ (from Z to Y in steps of X, stopping when Y is reached, or - for Real output - when the last digit of the answer at the SDIGS setting does not change anymore) ; terms must be decreasing ; returns number of iterations in Y ; interrupt by [EXIT] -- see Formula_SIGMAN.png : https://47calc.com/doc/Images/Formula_SIGMAN.png (TAM : $\omega \sum_n$ _ TamLbLOnly menu ; Info : <from> ENTER <to> ENTER <step> $\omega \sum_n$; displays Loop: n, current result, where n is iteration counter ; shown if flag MONIT is SET ; Ref : https://47calc.com/doc/Application%20Notes/AN0008_C47_JM_Programmed_Sums1.pdf)
	Command	Integer product (RPN)		$i \prod_n$	$i \prod_n$	Integer product using RPN program ; $i \prod_n$ (from Z to Y in steps of X) ; interrupt by [EXIT] -- see Formula_iPIn.png : https://47calc.com/doc/Images/Formula_iPIn.png (TAM : $i \prod_n$ _ TamLbLOnly menu ; Info : <from> ENTER <to> ENTER <step> $i \prod_n$; displays Loop: n, current result, where n is iteration counter ; shown if flag MONIT is SET)
	Command	Integer sum (RPN)		$i \sum_n$	$i \sum_n$	Integer sum using RPN program ; $i \sum_n$ (from Z to Y in steps of X) ; interrupt by [EXIT] -- see Formula_iSIGMAN.png : https://47calc.com/doc/Images/Formula_iSIGMAN.png (TAM : $i \sum_n$ _ TamLbLOnly menu ; Info : <from> ENTER <to> ENTER <step> $i \sum_n$; displays Loop: n, current result, where n is iteration counter ; shown if flag MONIT is SET)
	Command	Real product (RPN)		$\prod_n$	$\prod_n$	Real product using RPN program ; $\prod_n$ (from Z to Y in steps of X) ; interrupt by [EXIT] -- see Formula_PIn.png : https://47calc.com/doc/Images/Formula_PIn.png (TAM : $\prod_n$ _ TamLbLOnly menu ; Info : <from> ENTER <to> ENTER <step> $\prod_n$; displays Loop: n, current result, where n is iteration counter ; shown if flag MONIT is SET)
	Command	Solve cubic		SLVC	SLVC	Solve the cubic equation : $a_3 \times x^3 + a_2 \times x^2 + a_1 \times x + a_0 = 0$; parameters taken from 4 coefficients on the stack or from a 1x4 or 4x1 coefficient vector in X ; SLVC also solves linear equations for 1x2 or 2x1 input and quadratic for 1x3 or 3x1 ; tolerance set by SDIGS (Info : a_3 ENTER a_2 ENTER a_1 ENTER a_0 SLVC ; roots 1, 2, 3 in Z, Y, X ; for vector input, roots are returned in a vector ; $tzxy \rightarrow \vec{v}_4$ is available on menu [$\vec{v} \leftarrow \vec{v}_4$]F5 to create a 1x4 coefficient vector (also $yx \rightarrow \vec{v}_2$, $zyx \rightarrow \vec{v}_3$))
	Command	Solve polynomial		SLVP	SLVP	Solve the polynomial equation : $a_n \times x^n + a_{n-1} \times x^{n-1} + \dots + a_0 = 0$, $n \leq 11$; parameters taken from a 1 x n+1 or n+1 x 1 coefficient vector in X ; tolerance set by SDIGS (Ref : https://47calc.com/doc/Application%20Notes/AN0033_C47_R47_DE_Polynomial_roots.pdf)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Solve quadratic		SLVQ	SLVQ	Solve the quadratic equation : $a \times x^2 + b \times x + c = 0$; parameters taken from 3 coefficients on the stack or from a 1x3 or 3x1 coefficient vector in X ; SLVQ also solves linear equations for 1x2 or 2x1 input and cubic for 1x4 or 4x1 ; tolerance set by SDIGS (Info : a ENTER b ENTER c SLVQ ; roots 1, 2 in Y, X ; for vector input, roots are returned in a vector ; $zyx \rightarrow \vec{v}_3$ is available on menu $[\vec{v}_4 \rightarrow \vec{v}_1] F3$ to create a 1x3 coefficient vector (also $yx \rightarrow \vec{v}_2$, $tzyx \rightarrow \vec{v}_4$))
	Command	Specify RPN program for $\int f dx$		PGMINT	PGMINT	In Program Entry Mode, specify the RPN program (with MVAR variables) to be used as the expression, store values in variable(s), then open menu $\int f dx$ and select integrator and variable to be integrated for, integrator limits set from stack (Y, X) (TAM : PGMINT __ TamLbOnly menu ; Info : PGMINT can nest PGMINT inside : nesting is limited to 3, respectively 4 levels, on new models and simulator, but nesting is limited to 2 levels on DM42 hardware ; Error 66 -- Nesting too deep -- may be raised)
	Command	Specify RPN program for f' , f''	0	PGMDRV	PGMDRV	In Program Entry Mode, specify the RPN program (with MVAR variables) to be used as the function to be differentiated, store values in variable(s), then activate f' or f'' and specify variable to differentiate for and step size (TAM : PGMDRV __ TamLbOnly menu)
	Command	Specify RPN program for PLT f		PGMPLT	PGMPLT	In Program Entry Mode, specify the RPN program (with MVAR variables) to be used as the function to be plotted, store values in variable(s), then activate PLT f and specify variable to plot for, X-axis limits set from stack (Y, X) (TAM : PGMPLT __ TamLbOnly menu ; Info : PGMPLT can nest PGMINT and PGMSLV inside : nesting is limited to 3, respectively 4 levels, on new models and simulator, but plot nesting is disabled on DM42 hardware ; Error 66 -- Nesting too deep -- may be raised ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS)
	Command	Specify RPN program for SOLVE		PGMSLV	PGMSLV	In Program Entry Mode, 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, solver estimates set from stack (Y, X) ; tolerance set by SDIGS, but the solver converges to graph precision while (nested inside) plotting (TAM : PGMSLV __ TamLbOnly menu ; Info : PGMSLV can nest PGMINT and PGMSLV inside : nesting is limited to 3, respectively 4 levels, on new models and simulator, but nesting is limited to 2 levels on DM42 hardware ; Error 66 -- Nesting too deep -- may be raised ; when SDIGS is set to the maximum value of 34, plotting uses 15 instead, otherwise it uses the value set by SDIGS)
	Command	Sum (RPN)		$\sum_n$	$\sum_n$	Sum using RPN program ; $\sum_n$ (from Z to Y in steps of X) ; interrupt by [EXIT] -- see Formula_SIGMAN.png : https://47calc.com/doc/Images/Formula_SIGMAN.png (TAM : $\sum_n$ __ TamLbOnly menu ; Info : <from> ENTER <to> ENTER <step> $\sum_n$; displays Loop: n, current result, where n is iteration counter ; shown if flag MONIT is SET ; Ref : https://47calc.com/doc/Application%20Notes/AN0008_C47_IM_Programmed_Sums1.pdf)
Simulator	SIM-shift	Activate symbol input		SYM (AIM)		Activate symbol input when in Alpha Input Mode on simulator -- SYM (AIM)
Solver	Variable (real)	Lower estimate for solver	0	$\downarrow$ Est	$\downarrow$ EST	Reserved Real variable for the lower estimate for solver ($\downarrow$ Est) ; displays as $\downarrow$ E in menu ToolS for larger values ; set interactively from menu ToolS and set by realSlv $\downarrow$ or cpxSlv $\downarrow$ (Split screen : Displaying $\uparrow$ Est Upper = ; $\downarrow$ Est Lower =)
	Variable (real)	Upper estimate for solver	0	$\uparrow$ Est	$\uparrow$ EST	Reserved Real variable for the upper estimate for solver ($\uparrow$ Est) ; displays as $\uparrow$ E in menu ToolS for larger values ; set interactively from menu ToolS and set by realSlv $\uparrow$ or cpxSlv $\uparrow$ (Split screen : Displaying $\uparrow$ Est Upper = ; $\downarrow$ Est Lower =)
MENU		Solver [EQN]		[f Solve]	f Solve	Solver (variables) menu for the algebraic expression ($= 0$) entered in NEW ; enter value and press <var> 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 (Info : TI : "Select Solver Variable" is presented if ToolS is activated without a variable having been selected first) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_f_Solve_page_1.pdf
MENU		Solver tools		[ToolS]	ToolS	Solver tools and parameter settings (Ref : https://47calc.com/doc/C47/Reference/C47_Solver_Result_Codes.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ToolSlv_page_1.pdf
	Flag (system)	Solving	CLEAR	SOLVING		SET for solver running ; CLEAR for solver not running
	Command (nonpgm)	Complex solver (stack limits)		cpxSlv $\downarrow$	cpxSlv $\downarrow$	Solve the algebraic expression ($= 0$) entered in the equation editor for complex roots ; interrupt by [EXIT] ; tolerance set by SDIGS ; solver estimates set from stack (Y, X) (Info : Displays current evaluated number and iteration counter ; shown if flag MONIT is SET ; TI : Result Code = ; Accuracy $\approx$; <var> $_{prev}$ = ; <var> = (optionally: conjugates) ; Ref : https://47calc.com/doc/C47/Reference/C47_Solver_Result_Codes.pdf)
	Command (nonpgm)	Complex solver (variable limits)		cpxSlv	cpxSlv	Solve the algebraic expression ($= 0$) entered in the equation editor for complex roots ; interrupt by [EXIT] ; tolerance set by SDIGS ; solver estimates set from variables ($\downarrow$ Est, $\uparrow$ Est) (Info : Displays current evaluated number and iteration counter ; shown if flag MONIT is SET ; TI : Result Code = ; Accuracy $\approx$; <var> $_{prev}$ = ; <var> = (optionally: conjugates) ; Ref : https://47calc.com/doc/C47/Reference/C47_Solver_Result_Codes.pdf)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (nonpgm)	Real solver (stack limits)		realSlv _Y	realSlv _Y	Solve the algebraic expression (= 0) entered in the equation editor for real roots ; interrupt by [EXIT] ; tolerance set by SDIGS ; monitoring set by MONIT ; solver estimates set from stack (Y, X) (Info : Displays current evaluated number and iteration counter ; shown if flag MONIT is SET ; TI : Result Code = ; Accuracy ≈ ; <var>prev = ; <var> =)
	Command (nonpgm)	Real solver (variable limits)		realSlv	realSlv	Solve the algebraic expression (= 0) entered in the equation editor for real roots ; interrupt by [EXIT] ; tolerance set by SDIGS ; monitoring set by MONIT ; solver estimates set from variables (↓Est, ↑Est) (Info : Displays current evaluated number and iteration counter ; shown if flag MONIT is SET ; TI : Result Code = ; Accuracy ≈ ; <var>prev = ; <var> =)
Speed conversions	MENU	Speed conversions		[Speed:]	Speed:	Convert between units of speed (Info : Flag CONVP may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> → ← <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Speed_page_1.pdf
	Function (monadic, linked)	ft/s to km/h		ft/s→km/h		Convert foot per second to kilometre per hour (TI : km/h:)
	Function (monadic, linked)	ft/s to m/s		ft/s→m/s		Convert foot per second to metre per second (TI : m/s:)
	Function (monadic, linked)	ft/s to mph		ft/s→mph		Convert foot per second to mile per hour (TI : mph:)
	Function (monadic, linked)	fur/ftn to km/h		fur/ftn→km/h		Convert furlong per fortnight to kilometre per hour (TI : km/h:)
	Function (monadic, linked)	fur/ftn to m/s		fur/ftn→m/s		Convert furlong per fortnight to metre per second (TI : m/s:)
	Function (monadic, linked)	fur/ftn to mph		fur/ftn→mph		Convert furlong per fortnight to mile per hour (TI : mph:)
	Function (monadic, linked)	km/h to ft/s		km/h→ft/s		Convert kilometre per hour to foot per second (TI : ft/s:)
	Function (monadic, linked)	km/h to fur/ftn		km/h→fur/ftn		Convert kilometre per hour to furlong per fortnight (TI : fur/ftn: ; Info : single = ur/ftn←)
	Function (monadic, linked)	km/h to knot		km/h→knot		Convert kilometre per hour to knot (TI : knot:)
	Function (monadic, linked)	km/h to m/s		km/h→m/s		Convert kilometre per hour to metre per second (TI : m/s:)
	Function (monadic, linked)	km/h to mph		km/h→mph		Convert kilometre per hour to mile per hour (TI : mph:)
	Function (monadic, linked)	knot to km/h		knot→km/h		Convert knot to kilometre per hour (TI : km/h:)
	Function (monadic, linked)	knot to m/s		knot→m/s		Convert knot to metre per second (TI : m/s:)
	Function (monadic, linked)	knot to mph		knot→mph		Convert knot to mile per hour (TI : mph:)
	Function (monadic, linked)	m/s to ft/s		m/s→ft/s		Convert metre per second to foot per second (TI : ft/s:)
	Function (monadic, linked)	m/s to fur/ftn		m/s→fur/ftn		Convert metre per second to furlong per fortnight (TI : fur/ftn: ; Info : single = ur/ftn←)
	Function (monadic, linked)	m/s to km/h		m/s→km/h		Convert metre per second to kilometre per hour (TI : km/h:)
	Function (monadic, linked)	m/s to knot		m/s→knot		Convert metre per second to knot (TI : knot:)
	Function (monadic, linked)	m/s to mph		m/s→mph		Convert metre per second to mile per hour (TI : mph:)
	Function (monadic, linked)	mph to ft/s		mph→ft/s		Convert mile per hour to foot per second (TI : ft/s:)
	Function (monadic, linked)	mph to fur/ftn		mph→fur/ftn		Convert mile per hour to furlong per fortnight (TI : fur/ftn: ; Info : single = ur/ftn←)
	Function (monadic, linked)	mph to km/h		mph→km/h		Convert mile per hour to kilometre per hour (TI : km/h:)
	Function (monadic, linked)	mph to knot		mph→knot		Convert mile per hour to knot (TI : knot:)
	Function (monadic, linked)	mph to m/s		mph→m/s		Convert mile per hour to metre per second (TI : m/s:)
Stack	Setting (pgm)	Stack display lines	4	dSTACK	dSTACK ₄	Set the number of stack registers to be displayed (TAM : dSTACK ₄ TamNonReg menu)
	Setting (pgm)	Stack size 4	OFF	SSIZE4	SSIZE4 ○	Switch on 4 register stack (Info : SBI 8 ; 4 is shown if flag SBss is SET)
	Setting (pgm)	Stack Size 8	ON	SSIZE8	SSIZE8 ⊗	Switch on 8 register stack (Info : SBI 8 ; 4 is shown if flag SBss is SET)
	Setting	Entry RPN mode	ON	eRPN	eRPN ▣	Switch on interactive Entry RPN mode ; not programmable - in PEM, use flag eRPN (Info : If eRPN is ON, in Numeric Input Mode, ENTER pushes the input value onto the stack, and if OFF, then it also duplicates X to Y ; without input or in Program Entry Mode, ENTER always duplicates X to Y ; only if eRPN is OFF, automatic stack lift is disabled after ENTER ; stack size SBI (8 or 4) is displayed in normal font if eRPN is ON and inverted if OFF ; SBI 8 or 4 is shown if flag SBss is ON)
	MENU	Multi-stack commands		[MULTSTK]	MULTSTK	Multi-stack commands Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MULTSTK_page_1.pdf
	MENU	Stack		[STK]	STK	Stack functions ; Keyboard g[x]y Docs: https://47calc.com/doc/C47/Menus/C47_Menu_STK_page_1.pdf
	Function (monadic)	Add X into register		ST0+	ST0+	Add X into register or variable ; if X is a text string, the target is converted to text string and X is concatenated (Shortcut : [ST0] - [+]; TAM : ST0+ ₄ TamSto menu)
	Function (monadic)	Divide X into register		ST0÷	ST0÷	Divide register or variable by X (Shortcut : [ST0] - [÷]; TAM : ST0÷ ₄ TamSto menu)
	Function (monadic)	Multiply X into register		ST0×	ST0×	Multiply register or variable by X (Shortcut : [ST0] - [x]; TAM : ST0× ₄ TamSto menu)
	Function (monadic)	Recall and add		RCL+	RCL+	Recall register or variable and add X (Shortcut : [RCL] - [+]; TAM : RCL+ ₄ TamRcl menu)
	Function (monadic)	Recall and divide		RCL÷	RCL÷	Recall register or variable and divide by X (Shortcut : [RCL] - [÷]; TAM : RCL÷ ₄ TamRcl menu)
	Function (monadic)	Recall and multiply		RCL×	RCL×	Recall register or variable and multiply by X (Shortcut : [RCL] - [x]; TAM : RCL× ₄ TamRcl menu)
	Function (monadic)	Recall and subtract		RCL-	RCL-	Recall register or variable and subtract X (Shortcut : [RCL] - [-]; TAM : RCL- ₄ TamRcl menu)
	Function (monadic)	Subtract X from register		ST0-	ST0-	Subtract X from register or variable (Shortcut : [ST0] - [-]; TAM : ST0- ₄ TamSto menu)
	Flag (system)	Automatic Stack Lift	SET	ASLIFT		SET for automatic stack lift ; CLEAR for automatic stack lift inactive (Info : SBI is SIM only)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag	Entry RPN mode	SET	eRPN		SET for interactive Entry RPN mode; CLEAR for interactive Entry RPN mode inactive (Info : If eRPN is SET, in Numeric Input Mode, ENTER pushes the input value onto the stack, and if CLEAR, then it also duplicates X to Y ; without input or in Program Entry Mode, ENTER always duplicates X to Y ; only if eRPN is CLEAR, automatic stack lift is disabled after ENTER; stack size SBI (8 or 4) is displayed in normal font if eRPN is SET and inverted if CLEAR ; SBI 8 or 4 is shown if flag SBss is SET)
	Flag	Stack Size 8	SET	SSIZE8		SET for 8 register stack; CLEAR for 4 register stack (Info : SBI 8 ; 4 is shown if flag SBss is SET)
	Command (old)	Drop n stack registers		>nDROP<		Old version of the command nDROP that contained a bug when used in user programs (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use nDROP on [MULTSTK.1] fF3 instead)
	Command (old)	Duplicate n stack registers		>nDUP<		Old version of the command nDUP that contained a bug when used in user programs (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use nDUP on [MULTSTK.1] fF4 instead)
	Command (old)	Swap n stack registers		>nSWAP<		Old version of the command nSWAP that contained a bug when used in user programs (Info : May be found in older user programs and raises Error 56 -- Function has changed, please replace ; use nSWAP on [MULTSTK.1] fF5 instead)
	Command (nonpgm)	Undo		↶	UNDO	Restore complete stack, LASTx, STATS and system flags (Info : Error 128 -- Undo failed: likely no memory -- may be raised) ; Keyboard f[↶]
	Command	Drop n stack registers		nDROP	NDROP	Drop bottom n stack registers (n = 1-4) (TAM : nDROP _ TamNonReg menu)
	Command	Drop X		DROPx	DROPx	Drop X from stack (Shortcut : double [↶] (if <nDROP is CLEAR -- default))
	Command	Drop X, Y, Z		3DROP	3DROP	Drop X, Y, Z
	Command	Drop Y		DROPy	DROPy	Drop Y from stack
	Command	Duplicate n stack registers		nDUP	NDUP	Duplicate bottom n stack registers to the n stack registers above (n = 1-4) (TAM : nDUP _ TamNonReg menu)
	Command	Duplicate X, Y, Z to T, A, B		3DUP	3DUP	Duplicate X, Y, Z to T, A, B
	Command	Enter		ENTER	ENTER	If eRPN is SET, in Numeric Input Mode, ENTER pushes the input value onto the stack, and if CLEAR, then it also duplicates X to Y ; without input or in Program Entry Mode, ENTER always duplicates X to Y ; only if eRPN is CLEAR, automatic stack lift is disabled after ENTER (Info : Note that in Program Entry Mode, ENTER terminates numeric input without inserting an ENTER instruction (it is not necessary to separate numbers on consecutive program lines and - if eRPN is SET - ENTER would lead to unnecessary duplication of X) ; For assignments (ASN (f[1])), spell the name as 'ENTER' ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard [ENTER]
	Command	Fill stack		FILL	FILL	Fill stack with value in the X-register
	Command	Get Stack size	8	SSIZE#	SSIZE#	Get Stack size (Info : Set by SSIZE4 or SSIZE8 ; Flag SSIZE8 is default SET)
	Command	Last X		LASTx	LASTx	Recall last X (Register L) ; Keyboard f[xzy]
	Command	Recall		RCL	RCL	Recall value from register or variable (TAM : RCL _ TamRcl menu (TamRclTVM when menu TVM is open) ; TI : Rnn: or '<var>' : (or R.nn: or or '<var>' : ; Info : local registers .00-.nn etc. need to be allocated using LocR nn+1) ; Keyboard [RCL]
	Command	Recall maximum		RCL↑	RCL↑	Recall maximum of X and register or variable (Shortcut : [RCL] - [▲] ; TAM : RCL↑ _ TamRcl menu)
	Command	Recall minimum		RCL↓	RCL↓	Recall minimum of X and register or variable (Shortcut : [RCL] - [▼] ; TAM : RCL↓ _ TamRcl menu)
	Command	Recall stack		RCLS	RCLS	Recall complete stack from 4 or 8 registers (Shortcut : [RCL] - Stack ; TAM : RCLS _ TamRcl menu)
	Command	Recall X, Y		2RCL	2RCL	Recall X, Y from two consecutive registers, specify first register using TAM
	Command	Recall X, Y, Z		3RCL	3RCL	Recall X, Y, Z from three consecutive registers, specify first register using TAM (TAM : 3RCL _ Tam menu)
	Command	Roll down		R↓	R↓	Roll down stack ; Keyboard [R↓]
	Command	Roll up		R↑	R↑	Roll up stack
	Command	Shuffle stack		ℤ	ℤ	Shuffle stack: replace X Y Z T by selection of X Y Z T (TAM : ℤ _ TamShuffle menu)
	Command	Store		STO	STO	Store X in global register or variable and create variable if it does not exist ; maximum variable length is : 7 characters (TAM : STO _ TamSto menu (TamStoTVM when menu TVM is open) ; Info : local registers .00-.nn etc. need to be allocated using LocR nn+1 ; TI : Rnn: or '<var>' : (or R.nn: or or '<var>' : ; Ref : https://47calc.com/doc/C47/Reference/C47_Variable_naming.pdf) ; Keyboard [STO]
	Command	Store maximum		STO↑	STO↑	Store maximum of X and register or variable (Shortcut : [STO] - [▲] ; TAM : STO↑ _ TamSto menu)
	Command	Store minimum		STO↓	STO↓	Store minimum of X and register or variable (Shortcut : [STO] - [▼] ; TAM : STO↓ _ TamSto menu)
	Command	Store stack		STOS	STOS	Store entire stack in 4 or 8 registers (Shortcut : [STO] - Stack ; TAM : STOS _ TamSto menu)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Store X, Y		2STO	2STO	Store X, Y to two consecutive registers, specify first register using TAM
	Command	Store X, Y, Z		3STO	3STO	Store X, Y, Z to three consecutive registers, specify first register using TAM (TAM : 3STO __ Tam menu)
	Command	Swap n stack registers		nSWAP	NSWAP	Swap bottom n stack registers with the n stack registers above (n = 1-4) (TAM : nSWAP __ TamNonReg menu)
	Command	Swap T		$\frac{t}{z}$	$\frac{t}{z}$	Swap T and register (TAM : $\frac{t}{z}$ __ Tam menu)
	Command	Swap X		$\frac{x}{z}$	$\frac{x}{z}$	Swap X and register (TAM : $\frac{x}{z}$ __ Tam menu)
	Command	Swap X and Y		$\frac{x}{z}y$	$\frac{x}{z}y$	Swap register X and register Y ; Keyboard [$\frac{x}{z}y$]
	Command	Swap X, Y, Z and T, A, B		3SWAP	3SWAP	Swap X, Y, Z and T, A, B
	Command	Swap X, Y, Z, T and A, B, C, D		4SWAP	4SWAP	Swap X, Y, Z, T and A, B, C, D (0 :)
	Command	Swap Y		$\frac{y}{z}$	$\frac{y}{z}$	Swap Y and register (TAM : $\frac{y}{z}$ __ Tam menu)
	Command	Swap Z		$\frac{z}{z}$	$\frac{z}{z}$	Swap Z and register (TAM : $\frac{z}{z}$ __ Tam menu)
Stack and registers	Command	Show		SHOW	SHOW	Show item in maximum detail, according to data type (tag) : Long integers up to 10^{123} will be displayed in large numeric font ; up to 10^{416} in the medium standard font and up to 10^{1000} in tiny font ; Real numbers in SIG 6, UNIT 3, SCI 3 ; Complex numbers (RECT and POLAR) in SIG 4 ; Virtual compound data types are shown with all digits in large, medium or tiny font (Info : Key [R/S] changes view for Long integers ; Virtual compound data types (font size, next page) ; use [$\blacktriangle$] or [$\blacktriangledown$] to scroll through all registers (and matrix variables) ; when menu XFCNS is open, if XYZ contains result in Virtual compound data type, SHOW displays X : XY+Z = <value> and ditto for T : TA+8 and for the dedicated triple register storage, e.g. R90 : FMA: (fused multiply-add) ; Ref : https://47calc.com/doc/C47/Reference/C47_Datatypes.pdf) ; Keyboard f[.]
Status	SBI	Calculator busy	Blank	(Busy)		Calculator busy
	SBI	Program running	Blank	(Running)		Program running
	SBI	Program waiting	Blank	(Waiting)		Program waiting (user input required)
	Flag	Iteration monitoring	SET	MONIT		SET for displaying iteration status ; CLEAR for not displaying iteration status (Info : Pertains to executing programmable sums, solvers and integrator)
	Flag	Show angle symbol	CLEAR	SBang		SET for displaying angle symbol ($\angle$) in status bar ; CLEAR for not displaying angle symbol ($\angle$) in status bar (Info : Setting flag SBang also sets SBadm ; clearing SBadm also clears flag SBang)
	Flag	Show battery depletion	CLEAR	SBbatV		SET for displaying battery depletion (⎓) in status bar ; CLEAR for not displaying battery depletion (⎓) in status bar (Info : If SBbatV is CLEAR, the regular icon (⎓) will show, but only when the battery is low (flag LOWBAT is SET) ; when SBbatV is SET the special battery symbol has length proportional to 2.054 V to 3.045 V)
	Flag	Show complex mode	SET	SBcpx		SET for displaying Complex mode (L ; $\odot$) in status bar ; CLEAR for not displaying Complex mode (L ; $\odot$) in status bar
	Flag	Show complex results	CLEAR	SBcr		SET for displaying Complex results (C ; R) in status bar ; CLEAR for not displaying Complex results (C ; R) in status bar
	Flag	Show date	SET	SBdate		SET for displaying date (CLK ; YY) in status bar ; CLEAR for not displaying date (CLK ; YY) in status bar (Info : SBdate, SBtime and SBwoy are linked, only two out of three can be SET at most)
	Flag	Show degree symbol	SET	SBadm		SET for displaying degree symbol ($^\circ$) in status bar ; CLEAR for not displaying degree symbol ($^\circ$) in status bar (Info : Setting flag SBang also sets SBadm ; clearing SBadm also clears flag SBang)
	Flag	Show Fractions Mode and #BASE Mode	SET	SBfrac		SET for displaying Fractions Mode and #BASE Mode (a^b/n ; b/n ; $[+/-]$; $a^b/c \times I$; $b/c \times I$; $[+/-]$; #BASE ; #KEYA-F) in status bar ; CLEAR for not displaying Fractions Mode and #BASE Mode (a^b/n ; b/n ; $[+/-]$; $a^b/c \times I$; $b/c \times I$; $[+/-]$; #BASE ; #KEYA-F) in status bar
	Flag	Show integer mode	SET	SBint		SET for displaying integer mode ($\langle w \rangle ; \langle m \rangle$) in status bar ; CLEAR for not displaying integer mode ($\langle w \rangle ; \langle m \rangle$) in status bar (Info : $\langle w \rangle$ in SBI indicates word size (menu BITSET) ; $\langle m \rangle$ in SBI indicates sign mode (menu BITSET))
	Flag	Show matrix mode	CLEAR	SBmx		SET for displaying matrix mode (wrap ; grow) in status bar ; CLEAR for not displaying matrix mode (wrap ; grow) in status bar
	Flag	Show overflow/carry mode	SET	SBoc		SET for displaying overflow/carry mode ($^\circ$; c) in status bar ; CLEAR for not displaying overflow/carry mode ($^\circ$; c) in status bar
	Flag	Show printing	SET	SBprn		SET for displaying printing (⎓) in status bar ; CLEAR for not displaying printing (⎓) in status bar
	Flag	Show serial I/O	SET	SBser		SET for displaying serial I/O (⎓) in status bar ; CLEAR for not displaying serial I/O (⎓) in status bar
	Flag	Show shift on the right	CLEAR	SBshfR		SET for displaying shift on the right (f ; ⎓) in status bar ; CLEAR for displaying shift on the left (f ; ⎓) in status bar
	Flag	Show stack size	CLEAR	SBss		SET for displaying stack size (8 ; 4) in status bar ; CLEAR for not displaying stack size (8 ; 4) in status bar
	Flag	Show stopwatch running	SET	SBstw		SET for displaying stopwatch running (⌚) in status bar ; CLEAR for not displaying stopwatch running (⌚) in status bar

Category	Type	FullName	Default	Label	Catalog	Extended description
	Flag	Show time	CLEAR	SBtime		SET for displaying time (HH24:MM ; HH:MMam/pm) in status bar ; CLEAR for not displaying time (HH24:MM ; HH:MMam/pm) in status bar (Info : SBtime, SBdate and SBwoy are linked, only two out of three can be SET at most)
	Flag	Show TVM mode	SET	SBtvm		SET for displaying TVM mode (END ; BEG) in status bar ; CLEAR for not displaying TVM mode (END ; BEG) in status bar
	Flag	Show Week of Year and day number	CLEAR	SBwoy		SET for displaying Week of Year and day number (Ww#-d#) in status bar ; CLEAR for not displaying Week of Year and day number (Ww#-d#) in status bar (Info : SBwoy, SBdate and SBtime are linked, only two out of three can be SET at most)
	Command	View		VIEW	VIEW	View register or variable (with preceding TI) (TAM : VIEW __ Tam menu ; Info : if ACT is SET, VIEW will print when executed from the keyboard in MAN, NORM or TRACE ; in Program Entry Mode, EN must be SET as well ; if ACT is CLEAR and EN is SET, VIEW halts execution (R/S)) ; Keyboard f[0]
	Browser	Flag browser		FLGS	FLGS	Browse status pages (flags 00-99, lettered flags, status overview, system flags) (Info : Press [EXIT] to exit Flag browser and browse using [$\blacktriangle$] and [$\blacktriangledown$] ; browser STATUS in menu FLAG shows the same status pages in a different order ; Ref : https://47calc.com/doc/C47/Reference/C47_Flags_p1.pdf) ; Keyboard g[$\blacktriangledown$]
	Browser	Register browser		REGS	REGS	Browse all registers (Shortcut : [.] : Switch register/variable view,[R/S] : Switch contents/storage view,[RCL] : Recall bottom item,[$\blacktriangle$], [$\blacktriangledown$] , ABCDEGJKLMNO, 00..99 : Navigate, [EXIT] : Exit Register browser ; Ref : https://47calc.com/doc/C47/Reference/C47_Registers.pdf)
	Browser	Status		STATUS	STATUS	Show status pages (status overview, system flags, flags 00-99, lettered flags) (Info : Press [EXIT] to exit Status and browse using [$\blacktriangle$] and [$\blacktriangledown$] ; browser FLGS (g[$\blacktriangledown$]) shows the same status pages in a different order ; Ref : https://47calc.com/doc/C47/Reference/C47_Flags_p1.pdf)
Stopwatch	Shortcut (key)	Stopwatch key [.]		[.]		Keyboard shortcut for button LAP $\rightarrow$ R "Store lap time in register" ; Keyboard [.]
	Shortcut (key)	Stopwatch key [$\blacktriangleleft$]		[$\blacktriangleleft$]		Keyboard shortcut for button RESET "Reset stopwatch" ; Keyboard [$\blacktriangleleft$]
	Shortcut (key)	Stopwatch key [+]		[+]		Keyboard shortcut for button LAP $\rightarrow$ Σ "Add lap time to STATS" ; Keyboard [+]
	Shortcut (key)	Stopwatch key [ENTER]		[ENTER]		Keyboard shortcut for button TIM $\rightarrow$ R "Store time in register" ; Keyboard [ENTER]
	Shortcut (key)	Stopwatch key [R/S]		[R/S]		Keyboard shortcut for button R/S "Start/stop stopwatch" ; Keyboard [R/S]
	Shortcut (key)	Stopwatch key [Σ +]		[Σ +]		Keyboard shortcut for button TIM $\rightarrow$ Σ "Add time to STATS" ; Keyboard [Σ +]
	Setting (pgm)	Tenths of seconds	ON	0.1s		Show running time with tenths of seconds
	Flag (system)	Stopwatch running	CLEAR	RUNTIM		SET for stopwatch running ; CLEAR for stopwatch not running (Info : SBI O is shown if flag SBstw is SET)
	Command (nonpgm)	Recall and load to countdown		RCL Σ		Recall register value and load to countdown timer (decimal hours or time value)
	Command (nonpgm)	Recall and load to timer		RCL		Recall register value and load to stopwatch timer (decimal hours or time value) (Shortcut : [RCL] (STOPW active))
	Command	Add lap time to STATS	0	LAP $\rightarrow$ Σ		Add lap time to STATS (statistics matrix) and reset lap timer ; subscript shows number of data points (Shortcut : [+]) (STOPW active))
	Command	Add time to STATS	0	TIM $\rightarrow$ Σ		Add time to STATS (statistics matrix) ; subscript shows number of data points (Shortcut : [Σ +]) (STOPW active))
	Command	Reset stopwatch		RESET		Reset the stopwatch (Shortcut : [$\blacktriangleleft$] (STOPW active))
	Command	Start/stop stopwatch		R/S		Start/stop the stopwatch (Shortcut : [R/S] (STOPW active))
	Command	Store lap time in register	0	LAP $\rightarrow$ R		Store lap time in register, increment CRA (Current Register Address) and reset lap timer ; subscript shows CRA (Shortcut : [.] (STOPW active))
	Command	Store time in register	0	TIM $\rightarrow$ R		Store time in register and increment CRA (Current Register Address) ; subscript shows R/S (Shortcut : [ENTER] (STOPW active))
Summary statistics	MENU	Statistics		[STAT]	STAT	Statistics functions (Ref : https://47calc.com/doc/C47/Reference/C47_Formulas.pdf) ; Keyboard f[$\pm$] Docs : https://47calc.com/doc/C47/Menus/C47_Menu_STAT_page_1.pdf
	Command	Arithmetic means		$\bar{x}$	$\bar{x}$	Arithmetic means, x and y -- see Formula_XBAR.png : https://47calc.com/doc/Images/Formula_XBAR.png (TI : $\bar{x} = ; \bar{y} =$)
	Command	Geometric mean		$\bar{x}_G$	$\bar{x}_G$	Data evaluation of geometric mean -- see Formula_XG.png : https://47calc.com/doc/Images/Formula_XG.png (TI : $\bar{x}_G = ; \bar{y}_G =$)
	Command	Harmonic mean		$\bar{x}_H$	$\bar{x}_H$	Data evaluation of harmonic mean -- see Formula_XH.png : https://47calc.com/doc/Images/Formula_XH.png (TI : $\bar{x}_H = ; \bar{y}_H =$)
	Command	Interquartile range		x_{IQR}	x_{IQR}	Interquartile range for both X and Y ; this is equal to Q.3 - Q.1 (TI : $iqr_x = ; iqr_y =$)
	Command	Lower quartile		x_{Q1}	x_{Q1}	Lower quartile for both X and Y -- see https://en.wikipedia.org/wiki/Quartile#Method_1 (TI : $Q_1_x = ; Q_1_y =$)
	Command	Maxima		x_{MAX}	x_{MAX}	Maxima (TI : $x_{MAX} = ; y_{MAX} =$)
	Command	Median		x_{MEDN}	x_{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 (TI : $md_x = ; md_y =$)
	Command	Median absolute deviation		x_{MAD}	x_{MAD}	Median absolute deviation for both X and Y ; this is the median of the differences between each data point and the overall median (TI : $mad_x = ; mad_y =$)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command	Minima		x_{MIN}	x_{MIN}	Minima (TI : x_{MIN} = ; y_{MIN} =)
	Command	n		n	$n\Sigma$	Number of samples
	Command	Percentile		$x\%ILE$	$x\%ILE$	Percentile (TI : $pctile_x$ = ; $pctile_y$ =)
	Command	Population standard deviation		σ	σ	Population standard deviation -- see Formula_STDDEV.png : https://47calc.com/doc/Images/Formula_STDDEV.png (TI : σ_x = ; σ_y =)
	Command	Quadratic mean		$\bar{x}_{RMS}$	$\bar{x}_{RMS}$	Data evaluation of quadratic mean -- see Formula_XRMS.png : https://47calc.com/doc/Images/Formula_XRMS.png (TI : $\bar{x}_{RMS}$ = ; $\bar{x}_{RMS}$ =)
	Command	Sample standard deviation		s	s	Sample standard deviation -- see Formula_STDDEVWEIGHTED.png : https://47calc.com/doc/Images/Formula_STDDEVWEIGHTED.png (TI : s_x = ; s_y =)
	Command	Scattering factor for a lognormal population		ϵ_p	ϵ_p	Scattering factor for a lognormal population -- see Formula_SCATTFACtp.png : https://47calc.com/doc/Images/Formula_SCATTFACtp.png (TI : ϵ_{px} = ; ϵ_{py} =)
	Command	Scattering factor for a lognormal sample		ϵ	ϵ	Scattering factor for a lognormal sample -- see Formula_SCATTFACT.png : https://47calc.com/doc/Images/Formula_SCATTFACT.png (TI : ϵ_x = ; ϵ_y =)
	Command	Scattering factor of the geometric mean		ϵ_m	ϵ_m	Scattering factor of the geometric mean -- see Formula_SCATTFACTm.png : https://47calc.com/doc/Images/Formula_SCATTFACTm.png (TI : ϵ_{mx} = ; ϵ_{my} =)
	Command	Sigma-		Σ^-	Σ^-	Remove data from the statistics matrix (STATS) and put the values on the stack (TI : nnn data points ; Info : Σ^- pushes values onto the stack unless automatic stack lift is disabled, e.g. by Σ^+)
	Command	Sigma+		Σ^+	Σ^+	Enter data into the statistics matrix (STATS) ; the $[\Sigma^+]$ key also is the hotkey that can be used for special key assignments in NORMAL mode, using menu Σ^+ KEY (TI : nnn data point(s) ; Info : if a nx2 matrix is in X, Σ^+ appends the data to the STATS matrix (creating it as necessary) ; Σ^+ disables automatic stack lift for the next entry) ; Keyboard $[\Sigma^+]$
	Command	Standard error of the mean		s_m	s_m	Standard error of the mean -- see Formula_SM.png : https://47calc.com/doc/Images/Formula_SM.png (TI : s_{mx} = ; s_{my} =)
	Command	Standard error of the weighted mean		s_{mw}	s_{mw}	Standard error of the weighted mean -- see Formula_SMW.png : https://47calc.com/doc/Images/Formula_SMW.png (TI : s_{mw} =)
	Command	Stats range		x_{RANGE}	x_{RANGE}	Range for both X and Y ; this is equal to MAX - MIN of the statistics matrix (STATS) (TI : rg_x = ; rg_y =)
	Command	Upper quartile		x_{Q3}	x_{Q3}	Upper quartile for both X and Y -- see https://en.wikipedia.org/wiki/Quartile#Method_1 (TI : Q_3_x = ; Q_3_y =)
	Command	Weighted mean		$\bar{x}_w$	$\bar{x}_w$	Data evaluation of weighted mean -- see Formula_XW.png : https://47calc.com/doc/Images/Formula_XW.png (TI : $\bar{x}_w$ =)
	Command	Weighted population standard deviation		σ_w	σ_w	Weighted population standard deviation -- see Formula_STDDEVPOP.png : https://47calc.com/doc/Images/Formula_STDDEVPOP.png (TI : σ_w =)
	Command	Weighted population standard deviation		s_w	s_w	Weighted population standard deviation -- see Formula_SW.png : https://47calc.com/doc/Images/Formula_SW.png (TI : s_w =)
Summation	Command	Σx		Σx	Σx	Return Σx and Σy in X and Y respectively (TI : Σ_x = ; Σ_y =)
	Command	Summation of $\ln^2 x$		$\Sigma \ln^2 x$	$\Sigma \ln^2 x$	Summation of $\ln^2 x$ using statistics matrix (STATS)
	Command	Summation of $\ln^2 y$		$\Sigma \ln^2 y$	$\Sigma \ln^2 y$	Summation of $\ln^2 y$ using statistics matrix (STATS)
	Command	Summation of $\ln x$		$\Sigma \ln x$	$\Sigma \ln x$	Summation of $\ln x$ using statistics matrix (STATS)
	Command	Summation of $\ln x \cdot \ln y$		$\Sigma \ln x \cdot \ln y$	$\Sigma \ln x \cdot \ln y$	Summation of $\ln x \cdot \ln y$ using statistics matrix (STATS)
	Command	Summation of $\ln y$		$\Sigma \ln y$	$\Sigma \ln y$	Summation of $\ln y$ using statistics matrix (STATS)
	Command	Summation of x		Σx	Σx	Summation of x using statistics matrix (STATS)
	Command	Summation of x·lny		$\Sigma x \cdot \ln y$	$\Sigma x \cdot \ln y$	Summation of x·lny using statistics matrix (STATS)
	Command	Summation of x^2		Σx^2	Σx^2	Summation of x^2 using statistics matrix (STATS)
	Command	Summation of $x^2 \cdot \ln y$		$\Sigma x^2 \cdot \ln y$	$\Sigma x^2 \cdot \ln y$	Summation of $x^2 \cdot \ln y$ using statistics matrix (STATS)
	Command	Summation of $x^2 y$		$\Sigma x^2 y$	$\Sigma x^2 y$	Summation of $x^2 y$ using statistics matrix (STATS)
	Command	Summation of $x^2 y^{-1}$		$\Sigma x^2 y^{-1}$	$\Sigma x^2 y^{-1}$	Summation of $x^2 y^{-1}$ using statistics matrix (STATS)
	Command	Summation of x^3		Σx^3	Σx^3	Summation of x^3 using statistics matrix (STATS)
	Command	Summation of x^4		Σx^4	Σx^4	Summation of x^4 using statistics matrix (STATS)
	Command	Summation of x^{-1}		Σx^{-1}	Σx^{-1}	Summation of x^{-1} using statistics matrix (STATS)
	Command	Summation of $x^{-1} \cdot \ln y$		$\Sigma x^{-1} \cdot \ln y$	$\Sigma x^{-1} \cdot \ln y$	Summation of $x^{-1} \cdot \ln y$ using statistics matrix (STATS)
	Command	Summation of x^{-2}		Σx^{-2}	Σx^{-2}	Summation of x^{-2} using statistics matrix (STATS)
	Command	Summation of xy		Σxy	Σxy	Summation of xy using statistics matrix (STATS)
	Command	Summation of xy^{-1}		Σxy^{-1}	Σxy^{-1}	Summation of xy^{-1} using statistics matrix (STATS)
	Command	Summation of y		Σy	Σy	Summation of y using statistics matrix (STATS)
	Command	Summation of y·lnx		$\Sigma y \cdot \ln x$	$\Sigma y \cdot \ln x$	Summation of y·lnx using statistics matrix (STATS)
	Command	Summation of y^2		Σy^2	Σy^2	Summation of y^2 using statistics matrix (STATS)
	Command	Summation of y^{-1}		Σy^{-1}	Σy^{-1}	Summation of y^{-1} using statistics matrix (STATS)
	Command	Summation of y^{-2}		Σy^{-2}	Σy^{-2}	Summation of y^{-2} using statistics matrix (STATS)

Category	Type	FullName	Default	Label	Catalog	Extended description
System	MENU	Mode settings		[MODE]	MODE	System (mode settings) settings (Info : MODE replaces R47 menu PREF ; Ref : https://47calc.com/doc/Application%20Notes/AN0028_C47_R47_JM_Configurability_Tour.pdf) ; Keyboard f[+/-] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MODE_page_1.pdf
	MENU	Preferences		[PREF]	PREF	System preferences (Info : R47 menu PREF is replaced by MODE ; Ref : https://47calc.com/doc/Application%20Notes/AN0028_C47_R47_JM_Configurability_Tour.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PREF_page_1.pdf
	Flag (system)	Low Battery	CLEAR	LOWBAT		SET for low battery voltage; CLEAR for no low battery voltage (Info : SBI  is shown if flag SBbatV is SET)
	Flag (system)	USB Power	CLEAR	USB		SET for calculator running on USB power; CLEAR for calculator running on battery power
	Flag	Automatic shutdown	SET	AUTOFF		SET for automatic shutdown; CLEAR for automatic shutdown inactive (Info : Automatic shutdown delay is 10 m)
	Flag	Quiet	CLEAR	QUIET		SET for beeper disabled; CLEAR for beeper enabled (Info : Silences every sound source and clamps volume at both ends)
	Flag	Slow mode	CLEAR	SLOW		SET for slow mode; CLEAR for slow mode inactive (Info : Slow mode limits battery drain)
	Command (nonpgm)	Version		VERS	VERS	Show firmware version (TI : Firmware version)
	Command (nonpgm)	Who		WHO	WHO	Show calculator development team and contributor's names (TI : Team names)
	Command (HW)	Access DMCP menu		DMCP	DMCP	Access DMCP menu ; hardware only ; use EXIT to return without RESET -- see https://technical.swissmicros.com/dm42/doc/dm42_user_manual.html#dmcp_sys_menu (Info : Error 126 -- Not available on the simulator -- is raised when this command is performed on the simulator ; Ref : https://47calc.com/doc/C47/Reference/C47_DMCP_menu.pdf)
	Command (HW)	Activate USB disk		ActUSB	ActUSB	Activate USB disk without exiting to DMCP ; hardware only (Info : Error 126 -- Not available on the simulator -- is raised when this command is performed on the simulator)
	Command (HW)	Battery voltage		BATT	BATT	Get battery voltage ± 1 mV ; hardware only (TI : V = ; Info : Error 126 -- Not available on the simulator -- is raised when this command is performed on the simulator)
	Command (HW)	Disk state		DISK	DISK	Get DMCP status of the FAT disk ; hardware only (Info : Error 126 -- Not available on the simulator -- is raised when this command is performed on the simulator)
	Command	Last Ticks		LASTT	LASTT	Get number of ticks for the previous operation (command, function, program) (Info : One tick is 100 ms)
	Command	Memory (RAM)		MEM	MEM	Get amount of free RAM memory, in bytes (TI : Bytes =)
	Command	Off		OFF	OFF	Turn off calculator (Info : On the simulator, pressing OFF or quitting the application saves the calculator state in : backup.cfg) ; Keyboard f[EXIT]
	Command	Recall config		RCLCFG	RCLCFG	Recall config from register or variable (Shortcut : [RCL] - Config ; TAM : RCLCFG __ TamRcl menu ; Info : displays "Configuration data" in the stack)
	Command	Reset		RESET	RESET	Reset the calculator ; fill registers R10-R38 with special values ; this is the only way to do a clean restart if an autoloader file is present (YESN0 : Reset? ; Info : Clean startup, no autoloading of SAVFILES/C47auto.sav or SAVFILES/C47.sav ; does not reset the keyboard layout chosen in menu KEYS)
	Command	Store config		STOCFG	STOCFG	Store config in register or variable (Shortcut : [ST0] - Config ; TAM : STOCFG __ TamSto menu ; Info : displays "Configuration data" in the stack)
	Command	Switch on		<ON>		Switch on calculator (Shortcut : Switched OFF - [EXIT])
Temperature conversions	MENU	Temperature conversions		[Temp:]	Temp:	Convert between units of temperature (Info : Flag CONV _{HP} may be SET for <to>  <to> unit conversion buttons; CLEAR for <from>  <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Temp_page_1.pdf
	Function (monadic, linked)	°C to °F		°C→°F		Convert degrees Celsius to degrees Fahrenheit (TI : °F:)
	Function (monadic, linked)	°C to K		°C→K		Convert degrees Celsius to Kelvin (TI : K:)
	Function (monadic, linked)	°F to °C		°F→°C		Convert degrees Fahrenheit to degrees Celsius (TI : °C:)
	Function (monadic, linked)	°F to °R		°F→°R		Convert degrees Fahrenheit to degrees Rankine (TI : °R:)
	Function (monadic, linked)	°F to K		°F→K		Convert degrees Fahrenheit to Kelvin (TI : K:)
	Function (monadic, linked)	°R to °F		°R→°F		Convert degrees Rankine to degrees Fahrenheit (TI : °F:)
	Function (monadic, linked)	°R to K		°R→K		Convert degrees Rankine to Kelvin (TI : K:)
	Function (monadic, linked)	eV/k _B →K		eV/k _B →K		Convert electronvolt/Boltzmann constant to Kelvin (TI : K:)
	Function (monadic, linked)	K to °C		K→°C		Convert Kelvin to degrees Celsius (TI : °C:)
	Function (monadic, linked)	K to °F		K→°F		Convert Kelvin to degrees Fahrenheit (TI : °F:)
	Function (monadic, linked)	K to °R		K→°R		Convert Kelvin to degrees Rankine (TI : °R:)
	Function (monadic, linked)	K to eV/k _B		K→eV/k _B		Convert Kelvin to electronvolt/Boltzmann constant (TI : eV/k _B :)
Test programs	User Program	47DEFLT		47DEFLT		47DEFLT example program, available to load from PROGRAMS/47DEFLT.p47, using READP (Info : Restore typical settings on C47, R47, after HP35fun)
	User Program	AllOps		AllOps		AllOps test program, available to load from PROGRAMS/ALLPGMS/AllOps.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Bairsto		Bairsto		Bairsto test program, available to load from PROGRAMS/ALLPGMS/Bairsto.p47, using READP (after unzipping resources/testPgms.zip)

Category	Type	FullName	Default	Label	Catalog	Extended description
	User Program	BinetV3		BinetV3		BinetV3 example program, available to load from PROGRAMS/BinetV3.p47, using READP -- see https://forum.swissmicros.com/viewtopic.php?p=17761#p17761 (Info : Binet graphing, Jaco Mostert)
	User Program	ECA		ECA		ECA test program, available to load from PROGRAMS/ALLPGMS/ECA.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	FreeF		FreeF		FreeF test program, available to load from PROGRAMS/ALLPGMS/FreeF.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	FreeFp		FreeFp		FreeFp test program, available to load from PROGRAMS/ALLPGMS/FreeFp.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	FreFp2		FreFp2		FreFp2 test program, available to load from PROGRAMS/ALLPGMS/FreFp2.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	GudrmPL		GudrmPL		GudrmPL example program, available to load from PROGRAMS/GudrmPL.p47, using READP (Info : Gudermannian inverse function plot, J Mostert (jaymos))
	User Program	HP35fun		HP35fun		HP35fun example program, available to load from PROGRAMS/HP35fun.p47, using READP (Info : Replacement for this (development) function, being a fun HP35 mode. No emulation, no simulation. Just a bit of fun. Recommend to save your settings into register 35: [STO] - Config 35 ; Ref : https://47calc.com/doc/C47/Reference/C47_HP35.pdf)
	User Program	IBessI		IBessI		IBessI test program, available to load from PROGRAMS/ALLPGMS/IBessI.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	IBessP		IBessP		IBessP test program, available to load from PROGRAMS/ALLPGMS/IBessP.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	IntP		IntP		IntP test program, available to load from PROGRAMS/ALLPGMS/IntP.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Len2		Len2		Len2 test program, available to load from PROGRAMS/ALLPGMS/Len2.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Len3		Len3		Len3 test program, available to load from PROGRAMS/ALLPGMS/Len3.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Len4		Len4		Len4 test program, available to load from PROGRAMS/ALLPGMS/Len4.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Len5		Len5		Len5 test program, available to load from PROGRAMS/ALLPGMS/Len5.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Len6		Len6		Len6 test program, available to load from PROGRAMS/ALLPGMS/Len6.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Len7		Len7		Len7 test program, available to load from PROGRAMS/ALLPGMS/Len7.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Len8		Len8		Len8 test program, available to load from PROGRAMS/ALLPGMS/Len8.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	LISSAJ		LISSAJ		LISSAJ test program, available to load from PROGRAMS/ALLPGMS/LISSAJ.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	NQueens		NQueens		NQueens example program, available to load from PROGRAMS/NQueens.p47, using READP -- see https://forum.swissmicros.com/viewtopic.php?p=34743#p34743 (Info : NQueens implementation for test purposes; modified from Calcman54's program)
	User Program	OMp209		OMp209		OMp209 test program, available to load from PROGRAMS/ALLPGMS/OMp209.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	OMp212		OMp212		OMp212 test program, available to load from PROGRAMS/ALLPGMS/OMp212.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	OMp224		OMp224		OMp224 test program, available to load from PROGRAMS/ALLPGMS/OMp224.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	OMp226		OMp226		OMp226 test program, available to load from PROGRAMS/ALLPGMS/OMp226.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	OMp233		OMp233		OMp233 test program, available to load from PROGRAMS/ALLPGMS/OMp233.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	OMp235		OMp235		OMp235 test program, available to load from PROGRAMS/ALLPGMS/OMp235.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	OMp237		OMp237		OMp237 test program, available to load from PROGRAMS/ALLPGMS/OMp237.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	P1		P1		P1 Time value of money program, available to load from PROGRAMS/tvm/P1.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P10		P10		P10 Time value of money program, available to load from PROGRAMS/tvm/P10.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P11		P11		P11 Time value of money program, available to load from PROGRAMS/tvm/P11.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)

Category	Type	FullName	Default	Label	Catalog	Extended description
	User Program	P12		P12		P12 Time value of money program, available to load from PROGRAMS/tvm/P12.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P12a		P12a		P12a Time value of money program, available to load from PROGRAMS/tvm/P12a.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P1b		P1b		P1b Time value of money program, available to load from PROGRAMS/tvm/P1b.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P2		P2		P2 Time value of money program, available to load from PROGRAMS/tvm/P2.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P3		P3		P3 Time value of money program, available to load from PROGRAMS/tvm/P3.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P4		P4		P4 Time value of money program, available to load from PROGRAMS/tvm/P4.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P5		P5		P5 Time value of money program, available to load from PROGRAMS/tvm/P5.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P5b		P5b		P5b Time value of money program, available to load from PROGRAMS/tvm/P5b.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P6		P6		P6 Time value of money program, available to load from PROGRAMS/tvm/P6.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P6b		P6b		P6b Time value of money program, available to load from PROGRAMS/tvm/P6b.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P7		P7		P7 Time value of money program, available to load from PROGRAMS/tvm/P7.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P8		P8		P8 Time value of money program, available to load from PROGRAMS/tvm/P8.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	P9		P9		P9 Time value of money program, available to load from PROGRAMS/tvm/P9.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	Pall		Pall		Pall Time value of money program, available to load from PROGRAMS/tvm/Pall.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	PICTURE		PICTURE		PICTURE test program, available to load from PROGRAMS/ALLPGMS/PICTURE.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Prime		Prime		Prime test program, available to load from PROGRAMS/ALLPGMS/Prime.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	PROD		PROD		PROD test program, available to load from PROGRAMS/ALLPGMS/PROD.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	ReMp261		ReMp261		ReMp261 test program, available to load from PROGRAMS/ALLPGMS/ReMp261.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	SETT		SETT		SETT example program, available to load from PROGRAMS/SETT.p47, using READP -- see https://forum.swissmicros.com/viewtopic.php?p=33113#p33113 (Info : Example to effect bundled settings for different calculator setups)
	User Program	SlvCnst		SlvCnst		SlvCnst test program, available to load from PROGRAMS/ALLPGMS/SlvCnst.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	SlvExtr		SlvExtr		SlvExtr test program, available to load from PROGRAMS/ALLPGMS/SlvExtr.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	Speed		Speed		Speed test program, available to load from PROGRAMS/ALLPGMS/Speed.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	SPIRAL		SPIRAL		SPIRAL test program, available to load from PROGRAMS/ALLPGMS/SPIRAL.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	SPIRALK		SPIRALK		SPIRALK example program, available to load from PROGRAMS/SPIRALK.p47, using READP -- see https://forum.swissmicros.com/viewtopic.php?p=35422#p35422 (Info : Revised spiral program by Jaco Mostert, modified by RAprism, barrymead)
	User Program	STAT000		STAT000		STAT000 test program, available to load from PROGRAMS/ALLPGMS/STAT000.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	STAT001		STAT001		STAT001 test program, available to load from PROGRAMS/ALLPGMS/STAT001.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	STAT002		STAT002		STAT002 test program, available to load from PROGRAMS/ALLPGMS/STAT002.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	STAT111		STAT111		STAT111 test program, available to load from PROGRAMS/ALLPGMS/STAT111.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	STAT113		STAT113		STAT113 test program, available to load from PROGRAMS/ALLPGMS/STAT113.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	STAT115		STAT115		STAT115 test program, available to load from PROGRAMS/ALLPGMS/STAT115.p47, using READP (after unzipping resources/testPgms.zip)

Category	Type	FullName	Default	Label	Catalog	Extended description
	User Program	STATgr		STATgr		STATgr test program, available to load from PROGRAMS/ALLPGMS/STATgr.p47, using READP (after unzipping resources/testPgms.zip)
	User Program	TRiangle4		TRiangle4		TRiangle4 example program, available to load from PROGRAMS/TRiangle4.p47, using READP -- see https://forum.swissmicros.com/viewtopic.php?p=38646#p38646 (Info : Triangle program, original p58 42S Programming manual, ported by Bruce Young, Didier Lachieze, Barry Mead)
	User Program	TRivip13		TRivip13		TRivip13 example program, available to load from PROGRAMS/TRivip13.p47, using READP -- see https://forum.swissmicros.com/viewtopic.php?p=38746#p38746 (Info : Triangle program, re-imagined by Mike L)
	User Program	tstData		tstData		tstData program to store example data in registers 10-38, available to load from PROGRAMS/tstData.p47, using READP
	User Program	TT		TT		TT Time value of money program, available to load from PROGRAMS/tvm/TT.p47, using READP (Ref : https://47calc.com/doc/Application%20Notes/AN0012_C47_R47_JM_TVM.pdf)
	User Program	ΣV		ΣV		ΣV test program, available to load from PROGRAMS/ALLPGMS/SigmaV.p47, using READP (after unzipping resources/testPgms.zip)
Text functions	MENU	Alpha functions		[α.FN]	α.FN	Alpha (text string) functions ; Keyboard g[2] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_alpha-FN_page_1.pdf
	Function (monadic)	Alpha lowercase		αLOWER	αLOWER	Lowercase of text string in register or variable (TAM : αLOWER __ Tam menu)
	Function (monadic)	Alpha reverse		αREV	αREV	Reverse text string in register or variable (TAM : αREV __ Tam menu)
	Function (monadic)	Alpha rotate left		αRL	αRL	Rotate text string in register or variable by X characters to the left (TAM : αRL __ Tam menu ; Info : for empty text string, αRL acts as NOP)
	Function (monadic)	Alpha rotate right		αRR	αRR	Rotate text string in register or variable by X characters to the right (TAM : αRR __ Tam menu ; Info : for empty text string, αRR acts as NOP)
	Function (monadic)	Alpha shift left		αSL	αSL	Shift text string in register or variable to the left, deleting X characters (TAM : αSL __ Tam menu ; Info : for empty text string, αSL acts as NOP)
	Function (monadic)	Alpha shift right		αSR	αSR	Shift text string in register or variable to the right, deleting X characters (TAM : αSR __ Tam menu ; Info : for empty text string, αSR acts as NOP)
	Function (monadic)	Alpha to X		α→x	α→x	Convert first alpha character of text string in register or variable to character code (hexadecimal) and remove this character (TAM : α→x __ Tam menu ; Info : for empty text string, α→x returns 0 only)
	Function (monadic)	Alpha trim		αTRIM	αTRIM	Trim character(s) in X from both sides of text string in register or variable ; X may contain a character or character code (TAM : αTRIM __ Tam menu)
	Function (monadic)	Alpha UPPERcase		αUPPER	αUPPER	Uppercase of text string in register or variable (TAM : αUPPER __ Tam menu)
	Function (monadic)	X to alpha		x→α	x→α	Append character, character code (converted to character), text string or real matrix (converted to text string) to register or variable (TAM : x→α __ Tam menu ; Info : when X is selected as the destination register or variable, the input is replaced, not appended to (unless it contains a text string in which case it is appended to itself) ; in all cases the original value of X is saved in LASTx ; Error 33 -- Output would exceed 508 characters -- may be raised)
	Function (monadic, old)	X to alpha		>x→α<		Old version of the function x→α, that did not use TAM, to convert character code in X to alpha character (code value pushed to Y) (Info : May be found in older user programs, use x→α on [α.FN.1] F3 instead)
	Command (strike)	Alpha parse		αPARSE	αPARSE	Parse alpha input for numeric content (in development)
	Command	Alpha Integer Part		αIP	αIP	Append integer part of X to text string in register or variable (TAM : αIP __ Tam menu ; Info : if the register or variable does not contain a text string, αIP will first convert it to text string)
	Command	Alpha left		αLEFT	αLEFT	Left X characters of text string in register or variable ; X is dropped (TAM : αLEFT __ Tam menu)
	Command	Alpha length		αLENG	αLENG	Get length of text string in register or variable (TAM : αLENG __ Tam menu)
	Command	Alpha middle		αMID	αMID	Get X characters of text string, starting at position Y ; X and Y are dropped (TAM : αMID __ Tam menu)
	Command	Alpha position		αPOS	αPOS	Get position (starting from 0) of substring, character or character code in X within text string in register or variable (TAM : αPOS __ Tam menu ; Info : for substring not found or empty text string, αPOS returns -1)
	Command	Alpha right		αRIGHT	αRIGHT	Right X characters of text string in register or variable ; X is dropped (TAM : αRIGHT __ Tam menu)
	Browser (nonpgm)	Font browser		FBR	FBR	Browse system fonts (character tables) (Info : Press [EXIT] to exit Font browser and browse using [▲] and [▼] ; Ref : https://47calc.com/doc/C47/Reference/C47_Font_browser.pdf)
Text input	Setting (cyclic)	Normal-super-subscript	Normal	SUP SUB		Cycle normal-super-subscript (locking) (Shortcut : alpha g[▲] / alpha g[▼] ; Info : button shows SUP SUB ; αSUP ; αSUB)
	Setting	Alpha capslock	OFF	CAPS		Switch on alpha capslock ; not programmable (Shortcut : alpha f[▲] / alpha f[▼] ; Info : sets initial case of menu A...Ω/α...ω and αINTL/aintl; toggling alpha capslock clears Numlock)
	SBI	ASM active	Blank	(ASM active)		Alpha Selection Mode active

Category	Type	FullName	Default	Label	Catalog	Extended description
	SBI	ASM search	Blank	(ASM search)		Alpha Selection Mode two letter search
	MENU (item)	Alpha input		[α]	ALPHA	Open alpha menu to enter, edit and clear alpha input ; when TAM is active, open menu TamAlpha (Shortcut : longpress [XEQ] (even when M.14 is ON) ; Info : Alpha Input Mode is opened in UPPERCASE ; menu CAT [ALPHA] is available on keyboard alpha f[+] ; alpha g-shift characters can also be reached from alpha longpress keyboard) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_alpha_page_1.pdf
	MENU	Alpha input		[ALPHA]	ALPHA	Alpha menu to enter, edit and clear alpha input (Shortcut : alpha triple [f/g] (because HOME ^α is SET; not available in TamAlpha); alpha longer [f/g] (because HOME ^α is SET) ; Info : Alpha Input Mode is opened in UPPERCASE ; menu CAT [ALPHA] is available on keyboard alpha f[+] ; alpha g-shift characters can also be reached from alpha longpress keyboard) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_ALPHA_page_1.pdf
	Flag (system)	Alpha capslock	CLEAR	αCAP		SET for alpha capslock; CLEAR for alpha capslock inactive
	Flag	Alpha Entry	CLEAR	ALP.IN		SET for alpha entry; CLEAR for alpha entry inactive
	Flag	Alpha Input Mode	CLEAR	ALPHA		SET for alpha input mode; CLEAR for alpha input mode inactive
	Command	Assign		ASN	ASSIGN	Assign function, menu or character to a keyboard key or to a button in a menu ; for options including creating user defined menus, see reference page (TAM : ASSIGN α (f[XEQ]) activates "ASSIGN 'α TamAlpha menu 'α" and allows a function or command to be spelled, to assign it to a key or button ; Info : "ASSIGN 'α" indicates that a key or button can be pressed to assign it to another button or key ; use button ASN ([ALPHA.1]fF4) for character assignments in Alpha Input Mode ; this button is not shown in Program Entry Mode ; Ref : https://47calc.com/doc/C47/Reference/C47_Assignment.pdf) ; Keyboard f[1]
	Command	Clear alpha		CLA		In Alpha Input Mode, clear alpha input buffer ; in Equation Input Mode, clear equation (alpha) input buffer (Shortcut : alpha f[←]; alpha g[←]; alpha long [←]; EIM long [←])
	Command	Cursor to begin		↑		Jump to top left of alpha input ; Keyboard alpha [▲]
	Command	Cursor to end		↓		Jump to bottom right of alpha input ; Keyboard alpha [▼]
	Command	Edit X		X.EDIT	X.EDIT	Edit contents of stack register X as text (Shortcut : alpha long [ENTER]; EIM long [ENTER])
	Command	Swap X		X.SWAP	X.SWAP	Swap contents of X register with input of alpha buffer or equation editor
	Arrow	Cursor down		↓		Move cursor down
	Arrow	Cursor left		←		Move cursor left
	Arrow	Cursor left by 10		←		Move cursor left by 10 positions
	Arrow	Cursor right		→		Move cursor right
	Arrow	Cursor right by 10		→		Move cursor right by 10 positions
	Arrow	Cursor up		↑		Move cursor up
	Alpha-shift (special)	Flip case (one character)		<empty>		Set next alpha entry to flip lower/UPPER case (one character) (Shortcut : alpha f + <char>)
	Alpha-shift (special)	Set numeric (one digit)		<empty>		Set next alpha entry to numeric (one digit) or other alpha g-shifted character (Shortcut : alpha g + <char>)
	Alpha-shift	Alpha lock downwards		CASE DN		Move down alpha lock from N to A to a (Shortcut : alpha f[▼] ; Info : sets initial case of menu A...Ω/α...ω and αINTL/αintl)
	Alpha-shift	Alpha lock upwards		CASE UP		Move up alpha lock from A to a to N (Shortcut : alpha f[▲] ; Info : sets initial case of menu A...Ω/α...ω and αINTL/αintl)
	Alpha-shift	Subscript		↓		Subscript (Shortcut : alpha g[▼])
	Alpha-shift	Superscript		↑		Superscript (Shortcut : alpha g[▲])
Time conversions	Function (monadic, linked)	ftn to s		ftn→s		Convert fortnight to second (TI : s;)
	Function (monadic, linked)	s to ftn		s→ftn		Convert second to fortnight (TI : ftn;)
	Function (monadic, linked)	s to year		s→year		Convert second to year (TI : year;)
	Function (monadic, linked)	year to s		year→s		Convert year to second (TI : s;)
Time functions	Setting (pgm)	ISO Week of Year rule	ON	WOY _{ISO}	WOY _{ISO}	Set Week of Year rule to ISO ; week starts on Monday and week number depends on the first Thursday ; returns 1.4 (TI : Week of Year rule set: Monday.Thursday ; Info : SBI Wu#-d# is shown if flag SBwoy is SET)
	Setting (pgm)	Julian-Gregorian transition 1582		JG.1582	JG.1582	Set Julian-Gregorian transition date to 1582-10-15 (Info : First Gregorian day set: 1582-10-15)
	Setting (pgm)	Julian-Gregorian transition 1752		JG.1752	JG.1752	Set Julian-Gregorian transition date to 1752-09-14 (Info : First Gregorian day set: 1752-09-14)
	Setting (pgm)	Julian-Gregorian transition 1873		JG.1873	JG.1873	Set Julian-Gregorian transition date to 1873-01-01 (Info : First Gregorian day set: 1873-01-01)
	Setting (pgm)	Julian-Gregorian transition 1949		JG.1949	JG.1949	Set Julian-Gregorian transition date to 1949-10-01 (Info : First Gregorian day set: 1949-10-01)
	Setting (pgm)	Middle East Week of Year rule	OFF	WOY _{ME}	WOY _{ME}	Set Week of Year rule to Middle East ; week starts on Saturday and week number depends on the first Friday ; returns 6.5 (TI : Week of Year rule set: Saturday.Friday ; Info : SBI Wu#-d# is shown if flag SBwoy is SET)
	Setting (pgm)	Two digit year limit	0	YY	YY ₀	Set the limit <ccyy> for two digit year conversion in function .d (in NIM only), up to which the specified (tracked) century <cc> is implied and above which the prior century <cc-1> is used (TAM : YY ____ TamNonRegTrk menu ; Info : Displayed with subscript <ccyy> as YY _{ccyy})
	Setting (pgm)	US Week of Year rule	OFF	WOY _{US}	WOY _{US}	Set Week of Year rule to US ; week starts on Sunday and week number depends on the first Saturday ; returns 7.6 (TI : Week of Year rule set: Sunday.Saturday ; Info : SBI Wu#-d# is shown if flag SBwoy is SET)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Setting (pgm, old)	Two digit year limit	0	>YY<		Old version of the setting YY that could not use indirect addressing (Info : May be found in older user programs, use YY on [CLK.1]gF8 instead)
	Setting (pgm, HW)	Set date		SETDAT	SETDAT	Set date ; hardware only (DMCP setting) (Info : Error 126 -- Not available on the simulator -- is raised when this setting is performed on the simulator)
	Setting (pgm, HW)	Set time		SETTIM	SETTIM	Set time ; hardware only (DMCP setting) (Info : Error 126 -- Not available on the simulator -- is raised when this setting is performed on the simulator)
	MENU (item)	Date and time settings		[DTSET]	CLK#2	Activate menu CLK page 2 to access date and time settings (Info : Previews/OPENM as CLK#2) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_DTSET_page_1.pdf
	MENU	Clock		[CLK]	CLK	Clock functions, including setting date and time and julian day numbers (astronomy) (Info : Use TDISP to 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 ; Ref : https://47calc.com/doc/C47/Reference/C47_Date_formats.pdf) ; Keyboard g[5] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CLK_page_1.pdf
	Function (triadic)	Compose date		zyx→D	zyx→D	Compose date from Z, Y, X according to CLK date format settings DD.MM.YYYY, MM/DD/YYYY or YYYY-MM-DD ; implied conversion set by YY does not apply (TI : Weekday ; Ref : https://47calc.com/doc/C47/Reference/C47_Date_formats.pdf)
	Function (monadic)	Compose time		zyx→T	zyx→T	Compose time from 24H hours in Z, minutes in Y, seconds in X
	Function (monadic)	Convert decimal hours to time		H→T	H→T	Convert decimal hours to hh:mm:ss hours (Info : NIM input treated as sexagesimal (hh.mmss) format ; stack input treated as decimal value ; zero time (interval) displayed as "0.1s")
	Function (monadic)	Convert sexagesimal to time		H.MS→T	H.MS→T	Convert sexagesimal format input sequence to hh:mm:ss hours (Info : NIM input treated as sexagesimal (hh.mmss) format ; stack input treated as decimal value)
	Function (monadic)	Date to Julian day number		D→J	D→J	Convert date to Julian day number (JDN) (Info : Years are astronomical, 0 equals 1 BCE ; JDN .0 equals noon!)
	Function (monadic)	Date to X		D→x	D→x	Convert date to date input number according to CLK date format settings DD.MM.YYYY, MM/DD/YYYY or YYYY-MM-DD ; implied conversion set by YY does not apply (TI : yyyy.mmdd:)
	Function (monadic)	Day		DAY	DAY	Day (of date in X)
	Function (monadic)	Day of week number (ISO)		WDAY	WDAY	Day of week number (ISO, of date in X) regardless of Week of Year rule set (TI : [ISO day] Weekday)
	Function (monadic)	Decompose date		D→zyx	D→zyx	Decompose date to Z, Y, X according to CLK settings for date format
	Function (monadic)	Decompose time		T→zyx	T→zyx	Decompose time to 24H hours in Z, minutes in Y, seconds in X
	Function (monadic)	Hours		HOUR	HOUR	Hours (of time in X)
	Function (monadic)	Minutes		MINUTE	MINUTE	Minutes (of time in X)
	Function (monadic)	Month		MONTH	MONTH	Month (of date in X)
	Function (monadic)	Seconds		SECOND	SECOND	Seconds (of time in X)
	Function (monadic)	Set Julian-Gregorian transition	1752-09-14	J/G	J/G	Set the day that Julian date changes over to Gregorian date (using date in X) (Info : Input real number according to function x→D)
	Function (monadic)	Week of Year		WOY	WOY	Week of Year (of date in X) according to rule set (default : WOY _{ISO}) (TI : Week of Year =)
	Function (monadic)	X to date		x→D	x→D	Convert date input number to date according to CLK date format settings DD.MM.YYYY, MM/DD/YYYY or YYYY-MM-DD ; implied conversion set by YY does not apply (TI : Weekday ; Ref : https://47calc.com/doc/C47/Reference/C47_Date_formats.pdf)
	Function (monadic)	Year		YEAR	YEAR	Year (of date in X)
	Function (dyadic)	Date-time to Julian day number		DT→J	DT→J	Convert date, time in stack to Julian day number (JDN) (Info : Date, time can be in Y, X or X, Y ; Years are astronomical, 0 equals 1 BCE ; JDN .0 equals noon!)
	Command	Date		Date→D	Date→D	Current date (TI : Weekday (shown for dates in X, Y, Z))
	Command	Get Julian-Gregorian transition	1752-09-14	J/G#	J/G#	Get the day that Julian date changes over to Gregorian date (Info : Set by J/G)
	Command	Get Week of Year rule	1.4	WOY#	WOY#	Get Week of Year rule (default : WOY _{ISO} , week starts on Monday and week number depends on the first Thursday ; returns 1.4)
	Command	Julian day number to date-time		J→DT	J→DT	Convert Julian day number (JDN) to date in Y and time in X (Info : Years are astronomical, 0 equals 1 BCE ; JDN .0 equals noon!)
	Command	Time		Time→T	Time→T	Current time
	Command	Two digit year off		YYoff	YYoff	Switch off two digit year conversion in function .d (Info : Displayed in button for YY with subscript "off" as YY _{off})
	Command	Two digit year tracking		YYtrack	YYtrack	Switch on two digit year conversion tracking to imply the latest century <cc> used in function .d and set the limit for YY to <cc99> to use the full range of years (Info : Displayed in button for .d ; YY with subscript "T" as .d ; YY _{cc99t})
	App	Stopwatch		STOPW	STOPW	Stopwatch with running time, counter and countdown (playing BEEP after running down) ; Keyboard g[0]
Time/angle conversions	Function (monadic)	Sexagesimal		.ms ⁻¹	.ms ⁻¹	Convert hh:mm:ss hours or dd°mm'ss" degrees to sexagesimal format number (untag) (TI : hh/dd.mmss: ; Info : puts sexagesimal NIM input of .ms back as decimal number on the stack)
	Function (monadic, cyclic)	Minutes & seconds		.ms	.ms	Convert sexagesimal format input sequence or decimal stack value to hh:mm:ss hours or dd°mm'ss" degrees (cyclic) (Info : NIM input treated as sexagesimal (hh/dd.mmss) format ; stack input treated as decimal value) ; Keyboard g[√x]

Category	Type	FullName	Default	Label	Catalog	Extended description
Torque/moment conversions	MENU	Moment loads conversions		[MLoads:]	MLoads:	Convert between units of moment loads ; typical moment loads units (torque) (Info : Flag CONV _M may be SET for <to> <←> <to> unit conversion buttons ; CLEAR for <from> <←> <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MLoads_page_1.pdf
	Function (monadic, linked)	ft-lbf to N-m		ft-lbf→N-m		Convert foot-pound force to Newton-metre (TI : N-m)
	Function (monadic, linked)	in-lbf to N-m		in-lbf→N-m		Convert inch-pound force to Newton-metre (TI : N-m)
	Function (monadic, linked)	N-m to ft-lbf		N-m→ft-lbf		Convert Newton-metre to foot-pound force (TI : ft-lbf)
	Function (monadic, linked)	N-m to in-lbf		N-m→in-lbf		Convert Newton-metre to inch-pound force (TI : in-lbf)
Trailing input	Symbol	alpha button		α		Special alpha button on [TamLocalLabel.1] F3 to enter a local label name for LBL, GT0 or XEQ (TAM : LBL, GT0 or XEQ : _ TamAlpha menu ; the label will be entered as <label>)
	Symbol	Colon button		:		Special colon button on (e.g.) [TamLabel.1] F3 to create or specify a local named label for LBL, GT0 or XEQ (TAM : LBL, GT0 or XEQ : _ TamLocalLabel menu)
	MENU (TAM)	TAM (flag)		[FLG]		Presented in menu TamFlag to quickly select lettered flags (Info : Shortcut {A} indicates character key A (in TAM, etcetera ; Ref : https://47calc.com/doc/C47/Reference/C47_Flags_p1.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_FLG_page_1.pdf
	MENU (TAM)	TAM (register)		[REG]		Presented in TAM menus to quickly select lettered registers (Info : Shortcut {A} indicates character key A (in TAM, etcetera ; Ref : https://47calc.com/doc/C47/Reference/C47_Registers.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_REG_page_1.pdf
	MENU (TAM)	TAM mode		[TamMenu]		Transient Alpha Mode is activated for trailing input (menu selection) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection (→) activates TamIndirect ; TamMenu menu is typically activated by OPENM on [P.FN2.1] fF4) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamMenu_page_1.pdf
	MENU (TAM)	TAM mode (alpha)		[TamAlpha]		Transient Alpha Mode is activated for trailing input (alpha) (Shortcut : Activated from TAM menus by α (f[XEQ]) or TAM [XEQ] ; Info : Alpha Input Mode is opened in UPPERCASE for labels, lowercase for variables ; menu CAT [ALPHA] is available on keyboard alpha f[+] ; alpha g-shift characters can also be reached from alpha longpress keyboard) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamAlpha_page_1.pdf
	MENU (TAM)	TAM mode (compare)		[TamCmp]		Transient Alpha Mode is activated for trailing input (compare) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection (→) activates TamIndirect ; TamCmp menu is typically activated by x= ? on [TEST.1] F3) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamCmp_page_1.pdf
	MENU (TAM)	TAM mode (DMX)		[TamNonRegMax]		Transient Alpha Mode is activated for trailing input (DMX), allows selection of DMXmax (Info : Indirection (→) activates TamIndirect ; TamNonRegMax menu is typically activated by DMX on [DISP.1] fF4) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamNonRegMax_page_1.pdf
	MENU (TAM)	TAM mode (flag)		[TamFlag]		Transient Alpha Mode is activated for trailing input (flag) (Info : Indirection (→) activates TamIndirect ; TamFlag menu is typically activated by SF on [FLAG.1] F1) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamFlag_page_1.pdf
	MENU (TAM)	TAM mode (general)		[Tam]		Transient Alpha Mode is activated for trailing input (general) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection (→) activates TamIndirect ; Tam menu is typically activated by VIEW on f[0]) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Tam_page_1.pdf
	MENU (TAM)	TAM mode (indirect)		[TamIndirect]		Transient Alpha Mode is activated for trailing input (indirect) (Shortcut : Activated from TAM menus by Indirection (→) ; α (f[XEQ]) or TAM [XEQ] activates TamAlpha) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamIndirect_page_1.pdf
	MENU (TAM)	TAM mode (KEY)		[TamKey]		Transient Alpha Mode is activated for trailing input (KEYG, KEYX) (Info : TamKey menu is typically activated by) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamKey_page_1.pdf
	MENU (TAM)	TAM mode (label only)		[TamLbOnly]		Transient Alpha Mode is activated for trailing input (label only) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection (→) activates TamIndirect ; TamLbOnly menu is typically activated by PGMINT on [ADV.1] fF4) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamLbOnly_page_1.pdf
	MENU (TAM)	TAM mode (label)		[TamLabel]		Transient Alpha Mode is activated for trailing input (label) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection (→) activates TamIndirect ; Colon button (:) activates TamLocalLabel ; TamLabel menu is typically activated by LBL on g[LN] ; keyboard shortcuts {A}, {B}, {C}, {D}, {E}, {G}, {H}, {I}, {J}, {K}, {L} are available while menu TamLabel is open ([Σ+] = {A}, [1/x] = {B}, ..., [TAN] = {L})) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamLabel_page_1.pdf

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU (TAM)	TAM mode (local label)		[TamLocalLabel]		Transient Alpha Mode is activated for trailing input (local label) (Shortcut : Activated from TAM menus by Colon button (:) ; alpha button (α) activates TamAlpha) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamLocalLabel_page_1.pdf
	MENU (TAM)	TAM mode (norm)		[TamNorm]		Transient Alpha Mode is activated for trailing input (norm) (Info : Indirection ($\rightarrow$) activates TamIndirect ; TamNorm menu is typically activated by $\ x\ _p$ on [MATX.2] fF1) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamNorm_page_1.pdf
	MENU (TAM)	TAM mode (recall TVM)		[TamRclTVM]		Transient Alpha Mode is activated for trailing input (recall TVM) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection ($\rightarrow$) activates TamIndirect ; TamRclTVM menu is typically activated by [RCL] when menu TVM, AMORT or CASHFL is open) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamRclTVM_page_1.pdf
	MENU (TAM)	TAM mode (recall)		[TamRcl]		Transient Alpha Mode is activated for trailing input (recall) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection ($\rightarrow$) activates TamIndirect ; TamRcl menu is typically activated by [RCL]) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamRcl_page_1.pdf
	MENU (TAM)	TAM mode (shuffle)		[TamShuffle]		Transient Alpha Mode is activated for trailing input (shuffle) (Info : TamShuffle menu is typically activated by $\hat{z}$ on [STK.1] fF6) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamShuffle_page_1.pdf
	MENU (TAM)	TAM mode (store TVM)		[TamStoTVM]		Transient Alpha Mode is activated for trailing input (store TVM) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection ($\rightarrow$) activates TamIndirect ; TamStoTVM menu is typically activated by [STO] when menu TVM, AMORT or CASHFL is open) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamStoTVM_page_1.pdf
	MENU (TAM)	TAM mode (store)		[TamSto]		Transient Alpha Mode is activated for trailing input (store) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection ($\rightarrow$) activates TamIndirect ; TamSto menu is typically activated by [STO]) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamSto_page_1.pdf
	MENU (TAM)	TAM mode (track)		[TamNonRegTrk]		Transient Alpha Mode is activated for trailing input (two digit year and century tracking), allows setting YYtrack or YYoff (Info : Indirection ($\rightarrow$) activates TamIndirect ; TamNonRegTrk menu is typically activated by YY on [CLK.1] gF6) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamNonRegTrk_page_1.pdf
	MENU (TAM)	TAM mode (value)		[TamNonReg]		Transient Alpha Mode is activated for trailing input (value) (Info : Indirection ($\rightarrow$) activates TamIndirect ; TamNonReg menu is typically activated by DSP on [Prefix.1] fF6) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamNonReg_page_1.pdf
	MENU (TAM)	TAM mode (variable only)		[TamVarOnly]		Transient Alpha Mode is activated for trailing input (variable only) (Info : α (f[XEQ]) or TAM [XEQ] activates TamAlpha ; Indirection ($\rightarrow$) activates TamIndirect ; TamVarOnly menu is typically activated by MVAR on [P.FN2.1] F4) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TamVarOnly_page_1.pdf
	MENU (TAM, ASM)	TAM (program)		[PROG]	PROG	Presented in TAM menus for commands accessing global or local named labels (TAM : CAT.PROGS.* menu ; Info : Type characters 1-2 to search ; open menu PROG e.g. from button : on [TamLabel.1] F3 to activate menu TamLocalLabel, then menu PROG, to find local labels (if in scope)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_PROG_page_1.pdf
	MENU (ASM)	Menu		[MENU]	MENU	Presented in menu TamMenu for commands accessing menus (Info : Type characters 1-2 to search) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_MENU_page_1.pdf
	MENU (ASM)	VAR		[VAR]	VAR	Presented in TAM menus for commands accessing variables (TAM : CAT.VARS.* menu ; Info : Type characters 1-2 to search ; Matrix variables STATS and HISTO are also included in this menu, if created) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_VAR_page_1.pdf
	EasyAccess (TAM)	Flag A		A		Easy access to Flag A (104) (Shortcut : FF {A})
	EasyAccess (TAM)	Flag B		B		Easy access to Flag B (105) (Shortcut : FF {B})
	EasyAccess (TAM)	Flag C		C		Easy access to Flag C (106) (Shortcut : FF {C})
	EasyAccess (TAM)	Flag D		D		Easy access to Flag D (107) (Shortcut : FF {D})
	EasyAccess (TAM)	Flag E		E		Easy access to Flag E (217) (Shortcut : FF {E})
	EasyAccess (TAM)	Flag F		F		Easy access to Flag F (218)
	EasyAccess (TAM)	Flag G		G		Easy access to Flag G (219) (Shortcut : FF {G})
	EasyAccess (TAM)	Flag H		H		Easy access to Flag H (220) (Shortcut : FF {H})
	EasyAccess (TAM)	Flag I		I		Easy access to Flag I (109) (Shortcut : FF {I})
	EasyAccess (TAM)	Flag J		J		Easy access to Flag J (110) (Shortcut : FF {J})
	EasyAccess (TAM)	Flag K		K		Easy access to Flag K (111) (Shortcut : FF {K})
	EasyAccess (TAM)	Flag L		L		Easy access to Flag L (108) (Shortcut : FF {L})
	EasyAccess (TAM)	Flag M		M		Easy access to Flag M (211) (Shortcut : FF {M})
	EasyAccess (TAM)	Flag N		N		Easy access to Flag N (212) (Shortcut : FF {N})

Category	Type	FullName	Default	Label	Catalog	Extended description
	EasyAccess (TAM)	Flag 0		0		Easy access to Flag 0 (221) (Shortcut : FF {0})
	EasyAccess (TAM)	Flag P		P		Easy access to Flag P (213)
	EasyAccess (TAM)	Flag Q		Q		Easy access to Flag Q (214)
	EasyAccess (TAM)	Flag R		R		Easy access to Flag R (215)
	EasyAccess (TAM)	Flag S		S		Easy access to Flag S (216)
	EasyAccess (TAM)	Flag T		T		Easy access to Flag T (103)
	EasyAccess (TAM)	Flag U		U		Easy access to Flag U (222)
	EasyAccess (TAM)	Flag V		V		Easy access to Flag V (223)
	EasyAccess (TAM)	Flag W		W		Easy access to Flag W (224)
	EasyAccess (TAM)	Flag X		X		Easy access to Flag X (100)
	EasyAccess (TAM)	Flag Y		Y		Easy access to Flag Y (101)
	EasyAccess (TAM)	Flag Z		Z		Easy access to Flag Z (102)
	EasyAccess (TAM)	Infinity		∞		Special infinity button on [TamNorm.1]F2 to enter parameter value ∞
	EasyAccess (TAM)	Local label a		a		Easy access to local label a
	EasyAccess (TAM)	Local label A		A		Easy access to local label A (Shortcut : LBL {A})
	EasyAccess (TAM)	Local label b		b		Easy access to local label b
	EasyAccess (TAM)	Local label B		B		Easy access to local label B (Shortcut : LBL {B})
	EasyAccess (TAM)	Local label c		c		Easy access to local label c
	EasyAccess (TAM)	Local label C		C		Easy access to local label C (Shortcut : LBL {C})
	EasyAccess (TAM)	Local label d		d		Easy access to local label d
	EasyAccess (TAM)	Local label D		D		Easy access to local label D (Shortcut : LBL {D})
	EasyAccess (TAM)	Local label e		e		Easy access to local label e
	EasyAccess (TAM)	Local label E		E		Easy access to local label E (Shortcut : LBL {E})
	EasyAccess (TAM)	Local label f		f		Easy access to local label f
	EasyAccess (TAM)	Local label F		F		Easy access to local label F
	EasyAccess (TAM)	Local label g		g		Easy access to local label g
	EasyAccess (TAM)	Local label G		G		Easy access to local label G (Shortcut : LBL {G})
	EasyAccess (TAM)	Local label h		h		Easy access to local label h
	EasyAccess (TAM)	Local label H		H		Easy access to local label H (Shortcut : LBL {H})
	EasyAccess (TAM)	Local label i		i		Easy access to local label i
	EasyAccess (TAM)	Local label I		I		Easy access to local label I (Shortcut : LBL {I})
	EasyAccess (TAM)	Local label j		j		Easy access to local label j
	EasyAccess (TAM)	Local label J		J		Easy access to local label J (Shortcut : LBL {J})
	EasyAccess (TAM)	Local label k		k		Easy access to local label k
	EasyAccess (TAM)	Local label K		K		Easy access to local label K (Shortcut : LBL {K})
	EasyAccess (TAM)	Local label l		l		Easy access to local label l
	EasyAccess (TAM)	Local label L		L		Easy access to local label L (Shortcut : LBL {L})
	EasyAccess (TAM)	Register A		A		Easy access to Register A (104) : Stack level 5 (Shortcut : STO/RCL {A} ; TI : A:)
	EasyAccess (TAM)	Register B		B		Easy access to Register B (105) : Stack level 6 (Shortcut : STO/RCL {B} ; TI : B:)
	EasyAccess (TAM)	Register C		C		Easy access to Register C (106) : Stack level 7 (Shortcut : STO/RCL {C} ; TI : C:)
	EasyAccess (TAM)	Register D		D		Easy access to Register D (107) : Stack level 8 (Shortcut : STO/RCL {D} ; TI : D:)
	EasyAccess (TAM)	Register E		E		Easy access to Register E (217) : General purpose register E (Shortcut : STO/RCL {E} ; TI : E:)
	EasyAccess (TAM)	Register F		F		Easy access to Register F (218) : General purpose register F (TI : F:)
	EasyAccess (TAM)	Register G		G		Easy access to Register G (219) : General purpose register G ; also used to store third to last long integer prime factor for FACTORS (Shortcut : STO/RCL {G} ; TI : G:)
	EasyAccess (TAM)	Register H		H		Easy access to Register H (220) : General purpose register H ; also used to store second to last long integer prime factor for FACTORS (Shortcut : STO/RCL {H} ; TI : H:)
	EasyAccess (TAM)	Register I		I		Easy access to Register I (109) : General purpose register I ; also used to store current row of indexed matrix (Shortcut : STO/RCL {I} ; TI : I: or M[I, Jc]=<name> [<row> <col>] (if indexed matrix exists))
	EasyAccess (TAM)	Register J		J		Easy access to Register J (110) : General purpose register J ; also used to store current column of indexed matrix (Shortcut : STO/RCL {J} ; TI : J: or M[I, Jc]=<name> [<row> <col>] (if indexed matrix exists))
	EasyAccess (TAM)	Register K		K		Easy access to Register K (111) : General purpose register K ; also used in VarMNU to store variable name when MVAR-defined button (<var...>) is pressed without NIM input ; also used to store last long integer prime factor for FACTORS or the remaining number to be factorised ; also used as the default alpha register for commands compatible with the HP-42S/DM42(n) calculator (Shortcut : STO/RCL {K} ; TI : K:)
	EasyAccess (TAM)	Register L		L		Easy access to Register L (108) : Last X (Shortcut : STO/RCL {L} ; TI : L:)
	EasyAccess (TAM)	Register M		M		Easy access to Register M (211) : General purpose register M ; also used to store distribution parameter : mean, median, location, degree of freedom 1, centre, lower limit (Shortcut : STO/RCL {M} ; TI : M:)

Category	Type	FullName	Default	Label	Catalog	Extended description
	EasyAccess (TAM)	Register N		N		Easy access to Register N (212) : General purpose register N ; also used to store distribution parameter : degree of freedom 2, quantity, upper limit (Shortcut : STO/RCL {N} ; TI : N:)
	EasyAccess (TAM)	Register 0		0		Easy access to Register 0 (221) : General purpose register 0 (Shortcut : STO/RCL {0} ; TI : 0:)
	EasyAccess (TAM)	Register P		P		Easy access to Register P (213) : General purpose register P ; also used to store distribution parameter : probability (TI : P:)
	EasyAccess (TAM)	Register Q		Q		Easy access to Register Q (214) : General purpose register Q ; also used to store distribution parameter : shape, 3rd integer (TI : Q:)
	EasyAccess (TAM)	Register R		R		Easy access to Register R (215) : General purpose register R ; also used to store distribution parameter : rate (TI : R:)
	EasyAccess (TAM)	Register S		S		Easy access to Register S (216) : General purpose register S ; also used to store distribution parameter : scale, spread (TI : S:)
	EasyAccess (TAM)	Register T		T		Easy access to Register T (103) : Stack level 4 (TI : T:)
	EasyAccess (TAM)	Register U		U		Easy access to Register U (222) : General purpose register U (TI : U:)
	EasyAccess (TAM)	Register V		V		Easy access to Register V (223) : General purpose register V (TI : V:)
	EasyAccess (TAM)	Register W		W		Easy access to Register W (224) : General purpose register W (TI : W:)
	EasyAccess (TAM)	Register X		X		Easy access to Register X (100) : Stack level 1 (TI : X:)
	EasyAccess (TAM)	Register Y		Y		Easy access to Register Y (101) : Stack level 2 (TI : Y:)
	EasyAccess (TAM)	Register Z		Z		Easy access to Register Z (102) : Stack level 3 (TI : Z:)
	EasyAccess (TAM)	TVM annual interest rate	0	I/a		Easy access to variable I%/a to recall TVM annual interest rate (RCL 'I%/a') (TI : 'I%/a':)
	EasyAccess (TAM)	TVM annual interest rate	0	I/a		Easy access to variable I%/a to store TVM annual interest rate (STO 'I%/a') (TI : 'I%/a':)
	EasyAccess (TAM)	TVM compounding periods	12	cp/a		Easy access to variable CPER/a to recall TVM compounding periods (RCL 'CPER/a') (TI : 'CPER/a':)
	EasyAccess (TAM)	TVM compounding periods	12	cp/a		Easy access to variable CPER/a to store TVM compounding periods (STO 'CPER/a') (TI : 'CPER/a':)
	EasyAccess (TAM)	TVM future value	0	FV		Easy access to variable FV to recall TVM future value (RCL 'FV') (TI : 'FV':)
	EasyAccess (TAM)	TVM future value	0	FV		Easy access to variable FV to store TVM future value (STO 'FV') (TI : 'FV':)
	EasyAccess (TAM)	TVM number of payments	0	n		Easy access to variable NPPER to recall TVM number of payments (RCL 'NPPER') (TI : 'NPPER':)
	EasyAccess (TAM)	TVM number of payments	0	n		Easy access to variable NPPER to store TVM number of payments (STO 'NPPER') (TI : 'NPPER':)
	EasyAccess (TAM)	TVM payment	0	PMT		Easy access to variable PMT to recall TVM payment (RCL 'PMT') (TI : 'PMT':)
	EasyAccess (TAM)	TVM payment	0	PMT		Easy access to variable PMT to store TVM payment (STO 'PMT') (TI : 'PMT':)
	EasyAccess (TAM)	TVM payment period	12	pp/a		Easy access to variable PPER/a to recall TVM payment period (RCL 'PPER/a') (TI : 'PPER/a':)
	EasyAccess (TAM)	TVM payment period	12	pp/a		Easy access to variable PPER/a to store TVM payment period (STO 'PPER/a') (TI : 'PPER/a':)
	EasyAccess (TAM)	TVM present value	0	PV		Easy access to variable PV to recall TVM present value (RCL 'PV') (TI : 'PV':)
	EasyAccess (TAM)	TVM present value	0	PV		Easy access to variable PV to store TVM present value (STO 'PV') (TI : 'PV':)
	Command (TAM)	Index matrix		...INDEX	INDEX	Presented in menu TamSto to Index matrix in a register or variable ; set row I to 1, column J to 1 ; clear flags WRPEDG and WRPEND ; index wraps automatically, setting these flags ; index is dropped upon exit of matrix editor (TAM : INDEX __ Tam menu ; TI : M[I, Jc]=<name> [<row> <col>])
	Command (TAM)	Indirect T		→T		Presented in TAM menus for commands accessing indirect input using stack register T
	Command (TAM)	Indirect X		→X		Presented in TAM menus for commands accessing indirect input using stack register X
	Command (TAM)	Indirect Y		→Y		Presented in TAM menus for commands accessing indirect input using stack register Y
	Command (TAM)	Indirect Z		→Z		Presented in TAM menus for commands accessing indirect input using stack register Z
	Command (TAM)	Indirection		→		Presented in TAM menus for commands accessing indirect input (Info : Opens TamIndirect menu for the command that activated the current TAM menu)
	Command (TAM)	Real one		1.		Presented in menu TamCmp for comparisons (e.g. x< ?) to quickly enter 1.
	Command (TAM)	Real zero		0.		Presented in menu TamCmp for comparisons (e.g. x< ?) to quickly enter 0.
	Command (TAM)	Recall config		Config	RCLCFG	Presented in menu TamRcl to recall config from register or variable (Shortcut : [RCL] - Config ; TAM : RCLCFG __ TamRcl menu ; Info : displays "Configuration data" in the stack)
	Command (TAM)	Recall current matrix index		...IJ	RCLIJ	Presented in menu TamRcl to recall current matrix index (row I and column J) (Shortcut : [RCL] - ...IJ ; TI : M<name> :I,= ; M<name> :Jc=)
	Command (TAM)	Recall matrix element		...EL	RCLL	Presented in menu TamRcl to recall current element of indexed matrix (Shortcut : [RCL] - ...EL ; TI : M[I, Jc]=<name> [<row> <col>]=)
	Command (TAM)	Recall maximum		...Max	RCL↑	Presented in menu TamRcl to recall maximum of X and register or variable (Shortcut : [RCL] - [▲] ; TAM : RCL↑ __ TamRcl menu)
	Command (TAM)	Recall minimum		...Min	RCL↓	Presented in menu TamRcl to recall minimum of X and register or variable (Shortcut : [RCL] - [▼] ; TAM : RCL↓ __ TamRcl menu)
	Command (TAM)	Recall stack		Stack	RCLS	Presented in menu TamRcl to recall complete stack from 4 or 8 registers (Shortcut : [RCL] - Stack ; TAM : RCLS __ TamRcl menu)
	Command (TAM)	Recall vector element		...VEL _{nn}	RCLVEL	Presented in menu TamRcl to recall element n (TAM) of vector or matrix in X (Shortcut : [RCL] - ...VEL _{nn} ; TAM : RCLVEL __ Tam menu)
	Command (TAM)	Recall vector element 1		...VEL ₁	RCLVEL ₁	Presented in menu TamRcl to recall element 1 of vector or matrix in X
	Command (TAM)	Recall vector element 2		...VEL ₂	RCLVEL ₂	Presented in menu TamRcl to recall element 2 of vector or matrix in X
	Command (TAM)	Recall vector element 3		...VEL ₃	RCLVEL ₃	Presented in menu TamRcl to recall element 3 of vector or matrix in X

Category	Type	FullName	Default	Label	Catalog	Extended description
	Command (TAM)	Set current matrix index		...IJ	ST0IJ	Presented in menu TamSto to set current matrix index (row I and column J) (Shortcut : [ST0] - ...IJ ; TI : M<name> :I,= ; M<name> :J,=)
	Command (TAM)	Store config		Config	ST0CFG	Presented in menu TamSto to store config in register or variable (Shortcut : [ST0] - Config ; TAM : ST0CFG __ TamSto menu ; Info : displays "Configuration data" in the stack)
	Command (TAM)	Store maximum		...Max	ST0↑	Presented in menu TamSto to store maximum of X and register or variable (Shortcut : [ST0] - [▲] ; TAM : ST0↑ __ TamSto menu)
	Command (TAM)	Store minimum		...Min	ST0↓	Presented in menu TamSto to store minimum of X and register or variable (Shortcut : [ST0] - [▼] ; TAM : ST0↓ __ TamSto menu)
	Command (TAM)	Store stack		Stack	ST0S	Presented in menu TamSto to store entire stack in 4 or 8 registers (Shortcut : [ST0] - Stack ; TAM : ST0S __ TamSto menu)
	Command (TAM)	Store X into matrix element		...EL	ST0EL	Presented in menu TamSto to store X into current element of indexed matrix (Shortcut : [ST0] - Config ; TI : M[I,J]=<name> [<row> <col>]=)
	Command (TAM)	Store X into vector element		...VEL _{nn}	ST0VEL	Presented in menu TamSto to store X into element n (TAM) of vector or matrix in Y (Shortcut : [ST0] - ...VEL _{nn} ; TAM : ST0VEL ____ Tam menu)
	Command (TAM)	Store X into vector element 1		...VEL ₁	ST0VEL ₁	Presented in menu TamSto to store X into element 1 of vector or matrix in Y
	Command (TAM)	Store X into vector element 2		...VEL ₂	ST0VEL ₂	Presented in menu TamSto to store X into element 2 of vector or matrix in Y
	Command (TAM)	Store X into vector element 3		...VEL ₃	ST0VEL ₃	Presented in menu TamSto to store X into element 3 of vector or matrix in Y
	Command	Set DMX to maximum	64	DMXmax	DMXmax	Set DMX to maximum value (32500, stored as 0) (Info : SBI /max is shown if flag SBfrac is SET)
Trigonometry	MENU	More trig/hyperbolics		[TRG...]	TRG...	Extended trigonometry (and access to hyperbolic) functions Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TRG_MORE_page_1.pdf
	MENU	Trigonometry		[TRG]	TRG	Trigonometry functions (Info : TRG replaces R47 menu TRG_R47 ; Ref : https://47calc.com/doc/C47/Reference/C47_DMS-HMS.pdf) ; Keyboard g[+/-] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TRG_page_1.pdf
	MENU	Trigonometry		[TRG_R47]	TRG_R47	Trigonometry functions (Info : R47 menu TRG_R47 is replaced by TRG ; Ref : https://47calc.com/doc/C47/Reference/C47_DMS-HMS.pdf) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_TRG_R47_page_1.pdf
	Function (monadic)	Arc cosine		ACOS	ARCCOS	Inverse cosine (Info : Returns the principal value of the inverse cosine of X ; if flag CPXRES is SET, the domain is extended beyond [-1, +1] and complex results may be returned) ; Keyboard f[COS]
	Function (monadic)	Arc sine		ASIN	ARCSIN	Inverse sine (Info : Returns the principal value of the inverse sine of X ; if flag CPXRES is SET, the domain is extended beyond [-1, +1] and complex results may be returned) ; Keyboard f[SIN]
	Function (monadic)	Arc tangent		ATAN	ARCTAN	Inverse tangent (Info : Returns the principal value of the inverse tangent of X ; if flag SPCRES is SET, infinity input is allowed and the range is extended to [-90°, 90°] (inclusive)) ; Keyboard f[TAN]
	Function (monadic)	Cosine		COS	COS	Cosine ; Keyboard [COS]
	Function (monadic)	Sine		SIN	SIN	Sine ; Keyboard [SIN]
	Function (monadic)	Tangent		TAN	TAN	Tangent ; Keyboard [TAN]
	Function (monadic, legacy)	Arc cosine		>ACOS<		Legacy version of function ARCCOS : Inverse cosine (Info : Function >ACOS< is superceded by ARCCOS on f[COS])
	Function (monadic, legacy)	Arc sine		>ASIN<		Legacy version of function ARCSIN : Inverse sine (Info : Function >ASIN< is superceded by ARCSIN on f[SIN])
	Function (monadic, legacy)	Arc tangent		>ATAN<		Legacy version of function ARCTAN : Inverse tangent (Info : Function >ATAN< is superceded by ARCTAN on f[TAN])
	Function (dyadic)	Combinations		C _{yx}	COMB	Combinations of X out of Y
	Function (dyadic)	Expanded inverse tangent		ATAN2	ATAN2	Arc tangent of Y / X
Trigonometry and characters	Symbol	Cosine		COS		Cosine (Shortcut : alpha g[COS] ("COS"))
	Symbol	Sine		SIN		Sine (Shortcut : alpha g[SIN] ("SIN"))
	Symbol	Tangent		TAN		Tangent (Shortcut : alpha g[TAN] ("TAN"))
Unit conversions	MENU	Unit conversions		[CONV]	CONV	Unit conversions (Info : Flag CONV _{HP} may be SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> → <from> unit conversion buttons (default SET) ; Ref : https://47calc.com/doc/Application%20Notes/AN0024_C47_R47_JM_Assigning_custom_conversion_pairs.pdf) ; Keyboard f[5] Docs: https://47calc.com/doc/C47/Menus/C47_Menu_CONV_page_1.pdf
	Flag	Classic (HP) conversions	SET	CONV _{HP}		SET for <to> ↔ <to> unit conversion buttons ; CLEAR for <from> → <from> unit conversion buttons
	Flag	US or UK units	CLEAR	CONV _{US}		SET for US unit preference ; CLEAR for UK unit preference
User interface	Setting (pgm)	Bold font mode	OFF	BOLD	BOLD ☐	Switch on bold font mode
	Setting (pgm)	Function keys do full cycle	OFF	F.1234		Longpress control : Function keys do full cycle (Info : 1. primaryNM : unshifted ; 2. longpress : f-shift ; 3. longerpress : g-shift ; 4. longestpress : NOP)
	Setting (pgm)	Function keys skip f- and g-shift	OFF	F.14		Longpress control : Function keys skip f- and g-shift (Info : 1. primaryNM : unshifted ; 4. longestpress : NOP)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Setting (pgm)	Function keys skip g-shift	ON	F.12 ⁴		Longpress control : Function keys skip g-shift (Info : 1. primaryNM : unshifted ; 2. longpress : f-shift ; 4. longestpress : NOP)
	Setting (pgm)	Math keys do full cycle	ON	M.123 ⁴		Longpress control : Math keys do full cycle (Info : 1. primaryNM : unshifted ; 2. longpress : f-shift ; 3. longerpress : g-shift ; 4. longestpress : NOP ; Math keys are keys in row 2-4 except [E] and [←])
	Setting (pgm)	Math keys skip f- and g-shift	OFF	M.1 ⁴		Longpress control : Math keys skip f- and g-shift (Info : 1. primaryNM : unshifted ; 4. longestpress : NOP ; Math keys are keys in row 2-4 except [E] and [←])
	Setting (pgm)	Math keys skip g-shift	OFF	M.12 ⁴		Longpress control : Math keys skip g-shift (Info : 1. primaryNM : unshifted ; 2. longpress : f-shift ; 4. longestpress : NOP ; Math keys are keys in row 2-4 except [E] and [←])
	Setting (pgm)	Menu f,g-highlighting Full	ON	fg.FUL	fg.FUL ☒	Switch on menu f,g-highlighting full (Info : fg.FUL and fg.LIM are linked (both cannot be ON at the same time))
	Setting (pgm)	Menu f,g-highlighting Grey	OFF	fg.GR	fg.GR ☐	Switch on menu f,g-highlighting grey
	Setting (pgm)	Menu f,g-highlighting Limited	OFF	fg.LIM	fg.LIM ☐	Switch on menu f,g-highlighting limited (Info : fg.LIM and fg.FUL are linked (both cannot be ON at the same time))
	Setting (pgm)	Menu opens on p1	ON	MNUUp1	MNUUp1 ☒	Switch on menu opens on p1
	Setting	Double tap function keys	ON	g.2Tp	g.2Tp ☒	Switch on double tap function keys for g-shift ; not programmable - in PEM, use flag g.2Tp (Info : Blocked for navigation FN-keys (arrows) in editors)
	Setting	Longpress f and g shortcuts	ON	KEY.LP	KEY.LP ☒	Switch on longpress shortcuts for single shift keys ; not programmable - in PEM, use flag KEY.LP (Info : KEY.LP pertains to R47v0 f g layout and R47v2 ⁹ g layout only and also enables longpress assignment to single shift keys)
	Setting	Shift time-out of 4s	OFF	SH.4s	SH.4s ☐	Switch on shift time-out of 4s ; not programmable - in PEM, use flag SH.4s (Info : For layouts containing key [f/g] only)
	MENU (dialog)	Confirmation		[YESNO]		Confirmation menu (Info : YESNO menu is typically activated by RESET on [CLR.1] gF1) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_YESNO_page_1.pdf
	Flag	[←] executes DROPx or CLX	CLEAR	←DROP		SET for [←] executing DROPx (when not in NIM) ; CLEAR for [←] executing CLX (when not in NIM)
	Flag	Bold font mode	CLEAR	BOLD		SET for bold font mode ; CLEAR for regular font mode
	Flag	Double tap function keys	SET	g.2Tp		SET for double tap function keys for g-shift ; CLEAR for double tap function keys inactive (Info : Blocked for navigation FN-keys (arrows) in editors)
	Flag	Longpress f and g shortcuts	SET	KEY.LP		SET for longpress shortcuts for single shift keys ; CLEAR for longpress f and g shortcuts inactive (Info : KEY.LP pertains to R47v0 f g layout and R47v2 ⁹ g layout only and also enables longpress assignment to single shift keys)
	Flag	Menu f,g-highlighting Full	SET	fg.FUL		SET for menu f,g-highlighting full ; CLEAR for menu f,g-highlighting full inactive (Info : fg.FUL and fg.LIM are linked (both cannot be SET at the same time))
	Flag	Menu f,g-highlighting Grey	CLEAR	fg.GR		SET for menu f,g-highlighting grey ; CLEAR for menu f,g-highlighting grey inactive
	Flag	Menu f,g-highlighting Limited	CLEAR	fg.LIM		SET for menu f,g-highlighting limited ; CLEAR for menu f,g-highlighting limited inactive (Info : fg.LIM and fg.FUL are linked (both cannot be SET at the same time))
	Flag	Menu opens on p1	SET	MNUUp1		SET for menu opens on p1 ; CLEAR for menu opens on page last used
	Flag	Shift time-out of 4s	CLEAR	SH.4s		SET for shift time-out of 4s ; CLEAR for shift time-out of 4s inactive (Info : For layouts containing key [f/g] only)
	Command	Edit item		EDIT	EDIT	Edit item in X (any data type, except Config) in Normal Mode ; edit text strings, numbers or function parameters in the current step in Program Entry Mode ; when editing angles using EDIT, key .d removes the angle tag (Normal Mode and PEM) ; when editing short integers, key .d removes the base (Normal Mode and PEM) (Shortcut : longerpress [←] ; alpha longer [←] ; PEM long [←] ; EIM longer [←] ; Info : editing of long integers is limited to two input lines (about 99 digits depending on display settings) ; also note that for new models and simulator, the full set of features is supported (Complex numbers not yet implemented), but for DM42 hardware, numbers cannot be edited)
	Command	No		NO		Respond : No
	Command	Safe reset f/g timers		SRESET	SRESET	Safe reset, then toggle ON/OFF the timing related user interface options: fg.FUL ☒ , g.2Tp ☒ , SH.4s ☒ , HOME ⁹ ☒
	Command	Yes		YES		Respond : Yes
Vector operations	Setting (pgm)	Physics or Math	OFF	3DPHYS		Switch on Physics conventions in spherical polar vectors (Info : The notation convention is indicated by the TI subscript of the spherical polar vector : _p for Physics, _m for Math (determines ordering of vector elements, works on whole stack))
	Setting (pgm)	xyz or ijk vector display	OFF	3Dxyz		Switch on xyz (r8z) display of vectors (Info : Indicated by TI subscript of vector elements : _{xyz} or _{ijk})

Category	Type	FullName	Default	Label	Catalog	Extended description
	MENU	Vector conversions		$[\vec{v} \leftrightarrow \vec{v}]$	$\vec{v} \leftrightarrow \vec{v}$	Vector conversion functions (Ref : https://47calc.com/doc/C47/Reference/C47_Vector_display.pdf ; Info : Vectors are stored internally as rectangular and calculation results are presented in rectangular format ; to edit a vector, first put it in rectangular format ; vector TI's are shown for values in X only and a small font may be used ; vector TI's are minimised if the stack changes or [EXIT] is pressed (only ijk or rθk or pθφ is shown, without subscripts or brackets)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_VectorConv_page_1.pdf
	MENU	Vector menu		[VECTOR]	VECTOR	Vector functions (Ref : https://47calc.com/doc/C47/Reference/C47_Vector_display.pdf ; Info : Vectors are stored internally as rectangular and calculation results are presented in rectangular format ; to edit a vector, first put it in rectangular format ; vector TI's are shown for values in X only and a small font may be used ; vector TI's are minimised if the stack changes or [EXIT] is pressed (only ijk or rθk or pθφ is shown, without subscripts or brackets)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_VECTOR_page_1.pdf
	Function (triadic)	Compose 1x3 vector		$zyx \rightarrow \vec{v}_3$	$zyx \rightarrow \vec{v}_3$	Compose 1x3 vector from Z, Y, X, according to Physics or Math conventions (for spherical polar vectors) as set by flag 3DPHYS (default CLEAR) ; tagged angles in the input will be recognized and the appropriate coordinate tag will be applied to the vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ijk] [v ₁ v ₂ v ₃] ; spherical vector : [pθ; _i φ _k] _M [v ₁ v ₂ v ₃] ^s ; cylindrical vector : [rθ; _i k] [v ₁ v ₂ v ₃] ^c ; in Program Entry Mode, flag 3DPHYS determines whether $zyx \rightarrow \vec{v}_{3M}$ or $zyx \rightarrow \vec{v}_{3P}$ is stored ; notation convention is indicated by the TI subscript of the spherical polar vector : _P for Physics, _M for Math (determines ordering of vector elements, works on whole stack))
	Function (triadic)	Compose 1x3 vector (M)		$zyx \rightarrow \vec{v}_{3M}$	$zyx \rightarrow \vec{v}_{3M}$	Compose 1x3 vector from Z, Y, X, according to Math conventions (for spherical polar vectors) ; tagged angles in the input will be recognized and the appropriate coordinate tag will be applied to the vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ijk] [v ₁ v ₂ v ₃] ; spherical vector : [pθ; _i φ _k] _M [v ₁ v ₂ v ₃] ^s ; cylindrical vector : [rθ; _i k] [v ₁ v ₂ v ₃] ^c ; in Program Entry Mode, $zyx \rightarrow \vec{v}_{3M}$ is stored when $zyx \rightarrow \vec{v}_3$ is entered and flag 3DPHYS is CLEAR)
	Function (triadic)	Compose 1x3 vector (P)		$zyx \rightarrow \vec{v}_{3P}$	$zyx \rightarrow \vec{v}_{3P}$	Compose 1x3 vector from Z, Y, X, according to Physics conventions (for spherical polar vectors) ; tagged angles in the input will be recognized and the appropriate coordinate tag will be applied to the vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ijk] [v ₁ v ₂ v ₃] ; spherical vector : [pφ _k θ; _i] _P [v ₁ v ₂ v ₃] ^s ; cylindrical vector : [rθ; _i k] [v ₁ v ₂ v ₃] ^c ; in Program Entry Mode, $zyx \rightarrow \vec{v}_{3P}$ is stored when $zyx \rightarrow \vec{v}_3$ is entered and flag 3DPHYS is SET)
	Function (triadic, nonpgm)	Compose/decompose 1x3 vector		$zyx \leftrightarrow \vec{v}_3$	$zyx \leftrightarrow \vec{v}_3$	Compose/decompose 1x3 vector from/to Z, Y, X, according to Physics or Math conventions (for spherical polar vectors) as set by flag 3DPHYS (default CLEAR) ; tagged angles in the input will be recognized and the appropriate coordinate tag will be applied to the vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ijk] [v ₁ v ₂ v ₃] ; rectangular elements : [i] = ; [j] = ; [k] = ; spherical vector : [pθ; _i φ _k] _M [v ₁ v ₂ v ₃] ^s ; spherical elements : [p] = ; [θ _i] = ; [φ _k] = ; cylindrical vector : [rθ; _i k] [v ₁ v ₂ v ₃] ^c ; cylindrical elements : [r] = ; [θ _i] = ; [k] =)
	Function (tetradic, nonpgm)	Compose/decompose 1x4 vector		$tzyx \leftrightarrow \vec{v}_4$	$tzyx \leftrightarrow \vec{v}_4$	Compose/decompose 1x4 vector from/to T, Z, Y, X
	Function (tetradic)	Compose 1x4 vector		$tzyx \rightarrow \vec{v}_4$	$tzyx \rightarrow \vec{v}_4$	Compose 1x4 vector from T, Z, Y, X
	Function (monadic)	1x2 vector to Complex number		$\vec{v}_2 \rightarrow \mathbb{C}$	$\vec{v}_2 \rightarrow \mathbb{C}$	Convert 1x2 vector to Complex number (2.0-2.5)
	Function (monadic)	Complex to 1x2 vector		$\mathbb{C} \rightarrow \vec{v}_2$	$\mathbb{C} \rightarrow \vec{v}_2$	Convert Complex number to 1x2 vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ij] [v ₁ v ₂])
	Function (monadic)	Complex to or from 1x2 vector		$\mathbb{C} \leftrightarrow \vec{v}_2$	$\mathbb{C} \leftrightarrow \vec{v}_2$	Convert Complex number to or from 1x2 vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ij] [v ₁ v ₂])
	Function (monadic)	Decompose 1x2 vector		$\vec{v}_2 \rightarrow yx$	$\vec{v}_2 \rightarrow yx$	Decompose 1x2 vector to Y, X (Info : Display depends on flag 3Dxyz, for rectangular elements : [i] = ; [j] = ; polar elements : [r] = ; [θ _i] =)
	Function (monadic)	Decompose 1x3 vector		$\vec{v}_3 \rightarrow zyx$	$\vec{v}_3 \rightarrow zyx$	Decompose 1x3 vector to Z, Y, X, according to Physics or Math conventions (for spherical polar vectors) as set by flag 3DPHYS (default CLEAR) (Info : Display depends on flag 3Dxyz, for rectangular elements : [i] = ; [j] = ; [k] = ; spherical elements : [p] = ; [θ _i] = ; [φ _k] = ; cylindrical elements : [r] = ; [θ _i] = ; [k] = ; in Program Entry Mode, flag 3DPHYS determines whether $\vec{v}_3 \rightarrow zyx_M$ or $\vec{v}_3 \rightarrow zyx_P$ is stored)
	Function (monadic)	Decompose 1x3 vector (M)		$\vec{v}_3 \rightarrow zyx_M$	$\vec{v}_3 \rightarrow zyx_M$	Decompose 1x3 vector to Z, Y, X, according to Math conventions (for spherical polar vectors) (Info : Display depends on flag 3Dxyz, for rectangular elements : [i] = ; [j] = ; [k] = ; spherical elements : [p] = ; [θ _i] = ; [φ _k] = ; cylindrical elements : [r] = ; [θ _i] = ; [k] = ; in Program Entry Mode, $\vec{v}_3 \rightarrow zyx_M$ is stored when $\vec{v}_3 \rightarrow zyx$ is entered and flag 3DPHYS is CLEAR)

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic)	Decompose 1x3 vector (P)		$\vec{v}_3 \rightarrow \text{zyxp}$	$\vec{v}_3 \rightarrow \text{zyxp}$	Decompose 1x3 vector to Z, Y, X, according to Physics conventions (for spherical polar vectors) (Info : Display depends on flag 3Dxyz, for rectangular elements : [i] = ; [j] = ; [k] = ; spherical elements : [p] = ; [ϕ_k] = ; [θ_{ij}] = ; cylindrical elements : [r] = ; [θ_{ij}] = ; [k] = ; in Program Entry Mode, $\vec{v}_3 \rightarrow \text{zyxp}$ is stored when $\vec{v}_3 \rightarrow \text{zyx}$ is entered and flag 3DPHYS is SET)
	Function (monadic)	Decompose 1x4 vector		$\vec{v}_4 \rightarrow \text{tzyx}$	$\vec{v}_4 \rightarrow \text{tzyx}$	Decompose 1x4 vector to T, Z, Y, X
	Function (monadic)	Tag as Cylindrical		$\rightarrow \text{CYL}$	$\rightarrow \text{CYL}$	Tag a 3D vector as having cylindrical coordinates (Shortcut : $\rightarrow \text{P}$ cycles through $\rightarrow \text{SPH}$, then $\rightarrow \text{CYL}$; Info : display depends on flag 3Dxyz, for cylindrical vector : [$r\theta_{ij}k$] [$v_1 v_2 v_3$] ⁵)
	Function (monadic)	Tag as Spherical		$\rightarrow \text{SPH}$	$\rightarrow \text{SPH}$	Tag a 3D vector as having spherical coordinates, according to Physics or Math conventions (for spherical polar vectors) as set by flag 3DPHYS (default CLEAR) and indicated by the TI subscript of the spherical polar vector : p for Physics, m for Math (determines ordering of vector elements, works on whole stack) (Shortcut : $\rightarrow \text{P}$ cycles through $\rightarrow \text{SPH}$, then $\rightarrow \text{CYL}$; Info : display depends on flag 3DPHYS, for spherical vector : [$p\theta_{ij}\phi_k$] _m [$v_1 v_2 v_3$] ⁶)
	Function (dyadic)	Compose 1x2 vector		$\text{yx} \rightarrow \vec{v}_2$	$\text{yx} \rightarrow \vec{v}_2$	Compose 1x2 vector from Y, X ; tagged angles in the input will be recognized and the appropriate coordinate tag will be applied to the vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ij] [$v_1 v_2$] ; polar vector : [$r\theta_{ij}$] [$v_1 v_2$] ^p)
	Function (dyadic)	Perpendicular distance		$\ \vec{x}-\vec{y}\ _2$	$\ \vec{x}-\vec{y}\ _2$	Perpendicular distance between Real matrices or vectors ; Complex matrices or vectors
	Function (dyadic, nonpgm)	Compose/decompose 1x2 vector		$\text{yx} \leftrightarrow \vec{v}_2$	$\text{yx} \leftrightarrow \vec{v}_2$	Compose/decompose 1x2 vector from/to Y, X ; tagged angles in the input will be recognized and the appropriate coordinate tag will be applied to the vector (Info : Display depends on flag 3Dxyz, for rectangular vector : [ij] [$v_1 v_2$] ; rectangular elements : [i] = ; [j] = ; polar vector : [$r\theta_{ij}$] [$v_1 v_2$] ^p ; polar elements : [r] = ; [θ_{ij}] =)
	Flag	Physics or Math	CLEAR	3DPHYS		SET for Physics conventions in spherical polar vectors ; CLEAR for Math conventions in spherical polar vectors (Info : The notation convention is indicated by the TI subscript of the spherical polar vector : p for Physics, m for Math (determines ordering of vector elements, works on whole stack))
	Flag	xyz or ijk vector display	CLEAR	3Dxyz		SET for xyz (r0z) display of vectors ; CLEAR for ijk (r0k) display of vectors (Info : Indicated by TI subscript of vector elements : xyz or ijk)
	Command	Unit vector [0 0 1]		$\vec{v}_3[001]$	$\vec{v}_3[001]$	Create a new 3D unit vector with elements [0 0 1] (Info : Display depends on flag 3Dxyz, for rectangular vector : [ijk] [$v_1 v_2 v_3$])
	Command	Unit vector [0 1 0]		$\vec{v}_3[010]$	$\vec{v}_3[010]$	Create a new 3D unit vector with elements [0 1 0] (Info : Display depends on flag 3Dxyz, for rectangular vector : [ijk] [$v_1 v_2 v_3$])
	Command	Unit vector [0 1]		$\vec{v}_2[01]$	$\vec{v}_2[01]$	Create a new 2D unit vector with elements [0 1] (Info : Display depends on flag 3Dxyz, for rectangular vector : [ij] [$v_1 v_2$])
	Command	Unit vector [1 0 0]		$\vec{v}_3[100]$	$\vec{v}_3[100]$	Create a new 3D unit vector with elements [1 0 0] (Info : Display depends on flag 3Dxyz, for rectangular vector : [ijk] [$v_1 v_2 v_3$])
	Command	Unit vector [1 0]		$\vec{v}_2[10]$	$\vec{v}_2[10]$	Create a new 2D unit vector with elements [1 0] (Info : Display depends on flag 3Dxyz, for rectangular vector : [ij] [$v_1 v_2$])
Volume conversions	MENU	Volume conversions		[Volume:]	Volume:	Convert between units of volume ; this menu has two pages, one for the UK (imperial) units, the other for the US (customary) units : this is determined by flag CONVus, which is SET for US units on page 1 ; CLEAR for UK units on page 1 (default CLEAR) (Info : Flag CONVus may be SET for <to> $\leftrightarrow$ <to> unit conversion buttons ; CLEAR for <from> $\rightarrow$ <from> unit conversion buttons (default SET)) Docs: https://47calc.com/doc/C47/Menus/C47_Menu_Volume_page_1.pdf
	Function (monadic, linked)	barrel $\rightarrow$ m ³		barrel $\rightarrow$ m ³		Convert barrel to cubic metre (TI : m ³)
	Function (monadic, linked)	cup _{UK} $\rightarrow$ floz _{UK}		cup _{UK} $\rightarrow$ floz _{UK}		Convert UK imperial cup to UK fluid ounce (TI : floz _{UK})
	Function (monadic, linked)	cup _{UK} $\rightarrow$ mL		cup _{UK} $\rightarrow$ mL		Convert UK imperial cup to millilitre (TI : mL)
	Function (monadic, linked)	cup _{US} $\rightarrow$ floz _{US}		cup _{US} $\rightarrow$ floz _{US}		Convert US customary cup to US fluid ounce (TI : floz _{US})
	Function (monadic, linked)	cup _{US} $\rightarrow$ mL		cup _{US} $\rightarrow$ mL		Convert US customary cup to millilitre (TI : mL)
	Function (monadic, linked)	floz _{UK} $\rightarrow$ cup _{UK}		floz _{UK} $\rightarrow$ cup _{UK}		Convert UK fluid ounce to UK imperial cup (TI : cup _{UK})
	Function (monadic, linked)	floz _{UK} $\rightarrow$ gal _{UK}		floz _{UK} $\rightarrow$ gal _{UK}		Convert UK fluid ounce to UK gallon (TI : gal _{UK})
	Function (monadic, linked)	floz _{UK} $\rightarrow$ in ³		floz _{UK} $\rightarrow$ in ³		Convert UK fluid ounce to cubic inch (TI : in ³)
	Function (monadic, linked)	floz _{UK} $\rightarrow$ mL		floz _{UK} $\rightarrow$ mL		Convert UK fluid ounce to millilitre (TI : mL)
	Function (monadic, linked)	floz _{UK} $\rightarrow$ tbs _{UK}		floz _{UK} $\rightarrow$ tbs _{UK}		Convert UK fluid ounce to UK imperial tablespoon (TI : tbs _{UK})
	Function (monadic, linked)	floz _{UK} $\rightarrow$ tsp _{UK}		floz _{UK} $\rightarrow$ tsp _{UK}		Convert UK fluid ounce to UK imperial teaspoon (TI : tsp _{UK})
	Function (monadic, linked)	floz _{US} $\rightarrow$ cup _{US}		floz _{US} $\rightarrow$ cup _{US}		Convert US fluid ounce to US customary cup (TI : cup _{US})
	Function (monadic, linked)	floz _{US} $\rightarrow$ gal _{US}		floz _{US} $\rightarrow$ gal _{US}		Convert US fluid ounce to US gallon (TI : gal _{US})
	Function (monadic, linked)	floz _{US} $\rightarrow$ in ³		floz _{US} $\rightarrow$ in ³		Convert US fluid ounce to cubic inch (TI : in ³)
	Function (monadic, linked)	floz _{US} $\rightarrow$ mL		floz _{US} $\rightarrow$ mL		Convert US fluid ounce to millilitre (TI : mL)
	Function (monadic, linked)	floz _{US} $\rightarrow$ Tbsp _{US}		floz _{US} $\rightarrow$ Tbsp _{US}		Convert US fluid ounce to US customary tablespoon (TI : Tbsp _{US})
	Function (monadic, linked)	floz _{US} $\rightarrow$ tsp _{US}		floz _{US} $\rightarrow$ tsp _{US}		Convert US fluid ounce to US customary teaspoon (TI : tsp _{US})
	Function (monadic, linked)	ft ³ $\rightarrow$ gal _{UK}		ft ³ $\rightarrow$ gal _{UK}		Convert cubic foot to UK gallon (TI : gal _{UK})
	Function (monadic, linked)	ft ³ $\rightarrow$ gal _{US}		ft ³ $\rightarrow$ gal _{US}		Convert cubic foot to US gallon (TI : gal _{US})

Category	Type	FullName	Default	Label	Catalog	Extended description
	Function (monadic, linked)	$\text{ft}^3 \rightarrow \ell$		$\text{ft}^3 \rightarrow \ell$		Convert cubic foot to litre (TI : ℓ ;))
	Function (monadic, linked)	$\text{gal}_{\text{UK}} \rightarrow \text{fLoz}_{\text{UK}}$		$\text{gal}_{\text{UK}} \rightarrow \text{fLoz}_{\text{UK}}$		Convert UK gallon to UK fluid ounce (TI : fLoz_{UK} ;))
	Function (monadic, linked)	$\text{gal}_{\text{UK}} \rightarrow \text{ft}^3$		$\text{gal}_{\text{UK}} \rightarrow \text{ft}^3$		Convert UK gallon to cubic foot (TI : ft^3 ;))
	Function (monadic, linked)	$\text{gal}_{\text{UK}} \rightarrow \ell$		$\text{gal}_{\text{UK}} \rightarrow \ell$		Convert UK gallon to litre (TI : ℓ ;))
	Function (monadic, linked)	$\text{gal}_{\text{US}} \rightarrow \text{fLoz}_{\text{US}}$		$\text{gal}_{\text{US}} \rightarrow \text{fLoz}_{\text{US}}$		Convert US gallon to US fluid ounce (TI : fLoz_{US} ;))
	Function (monadic, linked)	$\text{gal}_{\text{US}} \rightarrow \text{ft}^3$		$\text{gal}_{\text{US}} \rightarrow \text{ft}^3$		Convert US gallon to cubic foot (TI : ft^3 ;))
	Function (monadic, linked)	$\text{gal}_{\text{US}} \rightarrow \ell$		$\text{gal}_{\text{US}} \rightarrow \ell$		Convert US gallon to litre (TI : ℓ ;))
	Function (monadic, linked)	$\text{in}^3 \rightarrow \text{fLoz}_{\text{UK}}$		$\text{in}^3 \rightarrow \text{fLoz}_{\text{UK}}$		Convert cubic inch to UK fluid ounce (TI : fLoz_{UK} ;))
	Function (monadic, linked)	$\text{in}^3 \rightarrow \text{fLoz}_{\text{US}}$		$\text{in}^3 \rightarrow \text{fLoz}_{\text{US}}$		Convert cubic inch to US fluid ounce (TI : fLoz_{US} ;))
	Function (monadic, linked)	$\text{in}^3 \rightarrow \text{m}\ell$		$\text{in}^3 \rightarrow \text{m}\ell$		Convert cubic inch to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{in}^3 \rightarrow \text{mm}^3$		$\text{in}^3 \rightarrow \text{mm}^3$		Convert cubic inch to cubic millimetre (TI : mm^3 ;))
	Function (monadic, linked)	$\ell \rightarrow \text{ft}^3$		$\ell \rightarrow \text{ft}^3$		Convert litre to cubic foot (TI : ft^3 ;))
	Function (monadic, linked)	$\ell \rightarrow \text{gal}_{\text{UK}}$		$\ell \rightarrow \text{gal}_{\text{UK}}$		Convert litre to UK gallon (TI : gal_{UK} ;))
	Function (monadic, linked)	$\ell \rightarrow \text{gal}_{\text{US}}$		$\ell \rightarrow \text{gal}_{\text{US}}$		Convert litre to US gallon (TI : gal_{US} ;))
	Function (monadic, linked)	$\ell \rightarrow \text{qt}_{\text{UK}}$		$\ell \rightarrow \text{qt}_{\text{UK}}$		Convert litre to UK quart (TI : qt_{UK} ;))
	Function (monadic, linked)	$\ell \rightarrow \text{qt}_{\text{US}}$		$\ell \rightarrow \text{qt}_{\text{US}}$		Convert litre to US quart (TI : qt_{US} ;))
	Function (monadic, linked)	$\text{m}^3 \rightarrow \text{barrel}$		$\text{m}^3 \rightarrow \text{barrel}$		Convert cubic metre to barrel (TI : barrel;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{cup}_{\text{UK}}$		$\text{m}\ell \rightarrow \text{cup}_{\text{UK}}$		Convert millilitre to UK imperial cup (TI : cup_{UK} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{cup}_{\text{US}}$		$\text{m}\ell \rightarrow \text{cup}_{\text{US}}$		Convert millilitre to US customary cup (TI : cup_{US} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{fLoz}_{\text{UK}}$		$\text{m}\ell \rightarrow \text{fLoz}_{\text{UK}}$		Convert millilitre to UK fluid ounce (TI : fLoz_{UK} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{fLoz}_{\text{US}}$		$\text{m}\ell \rightarrow \text{fLoz}_{\text{US}}$		Convert millilitre to US fluid ounce (TI : fLoz_{US} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{in}^3$		$\text{m}\ell \rightarrow \text{in}^3$		Convert millilitre to cubic inch (TI : in^3 ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{pint}_{\text{UK}}$		$\text{m}\ell \rightarrow \text{pint}_{\text{UK}}$		Convert millilitre to UK imperial pint (TI : pint_{UK} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{pint}_{\text{US}}$		$\text{m}\ell \rightarrow \text{pint}_{\text{US}}$		Convert millilitre to US liquid pint (TI : pint_{US} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{qt}_{\text{UK}}$		$\text{m}\ell \rightarrow \text{qt}_{\text{UK}}$		Convert millilitre to UK quart (TI : qt_{UK} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{qt}_{\text{US}}$		$\text{m}\ell \rightarrow \text{qt}_{\text{US}}$		Convert millilitre to US quart (TI : qt_{US} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{tbsp}_{\text{UK}}$		$\text{m}\ell \rightarrow \text{tbsp}_{\text{UK}}$		Convert millilitre to UK imperial tablespoon (TI : tbsp_{UK} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{Tbsp}_{\text{US}}$		$\text{m}\ell \rightarrow \text{Tbsp}_{\text{US}}$		Convert millilitre to US customary tablespoon (TI : Tbsp_{US} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{tsp}_{\text{UK}}$		$\text{m}\ell \rightarrow \text{tsp}_{\text{UK}}$		Convert millilitre to UK imperial teaspoon (TI : tsp_{UK} ;))
	Function (monadic, linked)	$\text{m}\ell \rightarrow \text{tsp}_{\text{US}}$		$\text{m}\ell \rightarrow \text{tsp}_{\text{US}}$		Convert millilitre to US customary teaspoon (TI : tsp_{US} ;))
	Function (monadic, linked)	$\text{mm}^3 \rightarrow \text{in}^3$		$\text{mm}^3 \rightarrow \text{in}^3$		Convert cubic millimetre to cubic inch (TI : in^3 ;))
	Function (monadic, linked)	$\text{pint}_{\text{UK}} \rightarrow \text{m}\ell$		$\text{pint}_{\text{UK}} \rightarrow \text{m}\ell$		Convert UK imperial pint to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{pint}_{\text{US}} \rightarrow \text{m}\ell$		$\text{pint}_{\text{US}} \rightarrow \text{m}\ell$		Convert US liquid pint to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{qt}_{\text{UK}} \rightarrow \ell$		$\text{qt}_{\text{UK}} \rightarrow \ell$		Convert UK quart to litre (TI : ℓ ;))
	Function (monadic, linked)	$\text{qt}_{\text{UK}} \rightarrow \text{m}\ell$		$\text{qt}_{\text{UK}} \rightarrow \text{m}\ell$		Convert UK quart to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{qt}_{\text{US}} \rightarrow \ell$		$\text{qt}_{\text{US}} \rightarrow \ell$		Convert US quart to litre (TI : ℓ ;))
	Function (monadic, linked)	$\text{qt}_{\text{US}} \rightarrow \text{m}\ell$		$\text{qt}_{\text{US}} \rightarrow \text{m}\ell$		Convert US quart to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{tbsp}_{\text{UK}} \rightarrow \text{fLoz}_{\text{UK}}$		$\text{tbsp}_{\text{UK}} \rightarrow \text{fLoz}_{\text{UK}}$		Convert UK imperial tablespoon to UK fluid ounce (TI : fLoz_{UK} ;))
	Function (monadic, linked)	$\text{tbsp}_{\text{UK}} \rightarrow \text{m}\ell$		$\text{tbsp}_{\text{UK}} \rightarrow \text{m}\ell$		Convert UK imperial tablespoon to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{Tbsp}_{\text{US}} \rightarrow \text{fLoz}_{\text{US}}$		$\text{Tbsp}_{\text{US}} \rightarrow \text{fLoz}_{\text{US}}$		Convert US customary tablespoon to US fluid ounce (TI : fLoz_{US} ;))
	Function (monadic, linked)	$\text{Tbsp}_{\text{US}} \rightarrow \text{m}\ell$		$\text{Tbsp}_{\text{US}} \rightarrow \text{m}\ell$		Convert US customary tablespoon to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{tsp}_{\text{UK}} \rightarrow \text{fLoz}_{\text{UK}}$		$\text{tsp}_{\text{UK}} \rightarrow \text{fLoz}_{\text{UK}}$		Convert UK imperial teaspoon to UK fluid ounce (TI : fLoz_{UK} ;))
	Function (monadic, linked)	$\text{tsp}_{\text{UK}} \rightarrow \text{m}\ell$		$\text{tsp}_{\text{UK}} \rightarrow \text{m}\ell$		Convert UK imperial teaspoon to millilitre (TI : $\text{m}\ell$;))
	Function (monadic, linked)	$\text{tsp}_{\text{US}} \rightarrow \text{fLoz}_{\text{US}}$		$\text{tsp}_{\text{US}} \rightarrow \text{fLoz}_{\text{US}}$		Convert US customary teaspoon to US fluid ounce (TI : fLoz_{US} ;))
	Function (monadic, linked)	$\text{tsp}_{\text{US}} \rightarrow \text{m}\ell$		$\text{tsp}_{\text{US}} \rightarrow \text{m}\ell$		Convert US customary teaspoon to millilitre (TI : $\text{m}\ell$;))