

R47v0 Ref page : Shortcuts ; also see : https://47calc.com/doc/Application%20Notes/AN0018_C47_R47_Keyboard.pdf

Label	Fullname	Type	Combo	Regular location	Description
-	Minus	Character	alpha g [-]	[αMath.1] gF5	Minus
;	Semicolon	Character	alpha f [.]	[αMisc.1] gF5	Character ;
:	Colon	Character	alpha f [R/S]	[αMisc.1] gF6	Character : ; in the equation editor, the colon can be used to name an expression (<name> : <expression>)
!	Exclamation mark	Character	alpha g [1/x]	[αMisc.2] F1	In edit equation, used as Factorial x! or Γ(x+1)
.	Full stop	Character	alpha g [.]	[αMisc.1] gF3	Character .
...EL	Store X into matrix element	Command (TAM)	[STO] - Config	[TamSto.1] gF1	Presented in menu TamSto to store X into current element of indexed matrix
...EL	Recall matrix element	Command (TAM)	[RCL] - ...EL	[TamRcl.1] gF1	Presented in menu TamRcl to recall current element of indexed matrix
...IJ	Set current matrix index	Command (TAM)	[STO] - ...IJ	[TamSto.1] gF2	Presented in menu TamSto to set current matrix index (row I and column J)
...IJ	Recall current matrix index	Command (TAM)	[RCL] - ...IJ	[TamRcl.1] gF2	Presented in menu TamRcl to recall current matrix index (row I and column J)
...Max	Store maximum	Command (TAM)	[STO] - [↑]	[TamSto.1] fF6	Presented in menu TamSto to store maximum of X and register or variable
...Max	Recall maximum	Command (TAM)	[RCL] - [↑]	[TamRcl.1] fF6	Presented in menu TamRcl to recall maximum of X and register or variable
...Min	Store minimum	Command (TAM)	[STO] - [↓]	[TamSto.1] fF5	Presented in menu TamSto to store minimum of X and register or variable
...Min	Recall minimum	Command (TAM)	[RCL] - [↓]	[TamRcl.1] fF5	Presented in menu TamRcl to recall minimum of X and register or variable
...VEL _{nn}	Store X into vector element	Command (TAM)	[STO] - ...VEL _{nn}	[TamSto.1] fF3	Presented in menu TamSto to store X into element n (TAM) of vector or matrix in Y
...VEL _{nn}	Recall vector element	Command (TAM)	[RCL] - ...VEL _{nn}	[TamRcl.1] fF3	Presented in menu TamRcl to recall element n (TAM) of vector or matrix in X
[M] ⁻¹	Invert matrix	Function (monadic)	[1/x]	[MATX.2] F1	Invert matrix in X (exists only when M ≠ 0)
/	Slash	Character	alpha g [R/S]	[αMath.1] gF2	Character /
#	Number sign	Character	alpha g [LN]	[αMisc.1] F2	Number sign, used for Short integer (8.02-8.16) base
^	Circumflex accent	Character	alpha g [y*]	[αMath.2] F1 (Exponent)	Raise to power
→CYL	Tag as Cylindrical	Function (monadic)	→P cycles through →SPH, then →CYL	[VECTOR.1] F3	Tag a 3D vector as having cylindrical coordinates
→SPH	Tag as Spherical	Function (monadic)	→P cycles through →SPH, then →CYL	[VECTOR.1] F2	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)
↑ (α ^{SUP})	Superscript	Alpha-shift	alpha g [↑]		Superscript
↓ (α _{SUB})	Subscript	Alpha-shift	alpha g [↓]		Subscript
↵	Carriage return	Character	alpha g [ENTER] ; alpha longer [ENTER] ; EIM longer [ENTER]	[αMisc.1] F1	Character ↵
↔	Left right arrow	Character	alpha g [x↔y]	[αMisc.3] F1	Character ↔
+	Plus	Character	alpha g [+]	[αMath.1] gF6	Plus
±	Plus-minus	Character	alpha g [CHS]	[αMath.2] gF3	Character ±
÷	Obelus	Character	alpha g [÷]	[αMath.1] gF3	Character ÷
×	Cross	Character	alpha g [x]	[αMath.1] gF4	Character ×
<f>lipchar	Flip case (one character)	Alpha-shift (special)	alpha f + <char>	alpha f [<f>]	Set next alpha entry to flip lower/UPPER case (one character)
<ON>	Switch on	Command	Switched OFF - [EXIT]		Switch on calculator
<E>	Exponent sign (AIM)	Character	alpha g [EEEX]	[αMath.4] F3	Character ε (displays as outline E in numeric font)
	Bar	Character	alpha g [STO]	[NEW (EQ_EDIT).2] fF5	Open or close absolute value
√	Square root	Character	alpha g [√x]	[NEW (EQ_EDIT).1] F4	Square root
A	Local label A	EasyAccess (TAM)	LBL {A}		Easy access to local label A
a b/c	Fractions modes	Setting (cyclic, pgm, stack)	<a> [.] [.] <c>	g [.]	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[y*]) ; starts in mode last used or as set by flags PROPFR, FRACT and IRFRAC ; When FRCYC is SET full cycle is available, when CLEAR, flag PROPFR is excluded (used as is) and OFF state is included in cycle
ALPHA	Alpha input	MENU	alpha long [<f>] (because KEY.LP is SET; also available in alpha mode of PEM; not available in TamAlpha)		Alpha menu to enter, edit and clear alpha input
π	Superscript pi	Character	alpha g [↑] π (alpha g [R↑])		Character π
B	Local label B	EasyAccess (TAM)	LBL {B}		Easy access to local label B
BIN	Binary	Setting (pgm)	#TAM [√x] (B)	[BASE.1] F4	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
BST	Backstep	Command (nonpgm)	[↑]	f [↑]	Backstep

Label	Fullname	Type	Combo	Regular location	Description
C	Local label C	EasyAccess (TAM)	LBL {C}		Easy access to local label C
CAPS	Alpha capslock	Setting	alpha f[+] / alpha f[+]	[ALPHA.1] gF1	Switch on alpha capslock ; not programmable
CASE DN	Alpha lock downwards	Alpha-shift	alpha f[+]		Move down alpha lock from N to A to a
CASE UP	Alpha lock upwards	Alpha-shift	alpha f[+]		Move up alpha lock from A to a to N
CLA	Clear alpha	Command	alpha f[+]; alpha g[+]; alpha long [+]; EIM long [+]		In Alpha Input Mode, clear alpha input buffer ; in Equation Input Mode, clear equation (alpha) input buffer
CLN	Clear number	Command	NIM long [+]		Clear numeric input buffer
CLRMOD	Clear modes	Command	longerpress [EXIT]; NIM longer [EXIT]; alpha longer [EXIT]; PEM longest [EXIT]; EIM longer [EXIT]	[CLR.1] gF2	Clear all menus, entry, fraction and base modes
CLSTK	Clear stack	Command	longpress [+]	[CLR.1] F6	Clear stack registers : when stack size is 8, clears XYZT ABCD ; when stack size is 4, clears XYZT ; if setting iLI is ON, the stack is filled with 0 (Long integers)
Config	Store config	Command (TAM)	[STO] - Config	[TamSto.1] fF1	Presented in menu TamSto to store config in register or variable
Config	Recall config	Command (TAM)	[RCL] - Config	[TamRcl.1] fF1	Presented in menu TamRcl to recall config from register or variable
D	Local label D	EasyAccess (TAM)	LBL {D}	[TamLabel.1] F4	Easy access to local label D
DEC	Decimal (short integer)	Setting (pgm)	#TAM [y*] (D)	[BASE.1] F2	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
DEC	Decimal (short integer)	Setting (pgm)	#TAM [ENTER]	[BASE.1] F2	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
di<g>it	Set numeric (one digit)	Alpha-shift (special)	alpha g[<g>]		Set next alpha entry to numeric (one digit) or other alpha g-shifted character
DR0Px	Drop X	Command	double [+]	[STK.1] F1	Drop X from stack
E	Local label E	EasyAccess (TAM)	LBL {E}	[TamLabel.1] F5	Easy access to local label E
e	eEuler	Character	alpha g[LOG]	[αMath.3] gF4	Character e
EDIT	Edit item	Command	longerpress [+]; alpha longer [+]; PEM long [+]; EIM longer [+]	[P.FN.1] gF1	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)
F	Local label F	EasyAccess (TAM)	LBL {F}	[TamLabel.1] F6	Easy access to local label F
Flag A	Flag A	EasyAccess (TAM)	FF {A}	[FLG.1] fF3	Easy access to Flag A (104)
Flag B	Flag B	EasyAccess (TAM)	FF {B}	[FLG.1] fF4	Easy access to Flag B (105)
Flag C	Flag C	EasyAccess (TAM)	FF {C}	[FLG.1] fF5	Easy access to Flag C (106)
Flag D	Flag D	EasyAccess (TAM)	FF {D}	[FLG.1] fF6	Easy access to Flag D (107)
Flag E	Flag E	EasyAccess (TAM)	FF {E}	[FLG.2] F1	Easy access to Flag E (217)
Flag F	Flag F	EasyAccess (TAM)	FF {F}	[FLG.2] F2	Easy access to Flag F (218)
Flag G	Flag G	EasyAccess (TAM)	FF {G}	[FLG.2] F3	Easy access to Flag G (219)
Flag H	Flag H	EasyAccess (TAM)	FF {H}	[FLG.2] F4	Easy access to Flag H (220)
Flag I	Flag I	EasyAccess (TAM)	FF {I}	[FLG.1] gF1	Easy access to Flag I (109)
Flag J	Flag J	EasyAccess (TAM)	FF {J}	[FLG.1] gF2	Easy access to Flag J (110)
Flag K	Flag K	EasyAccess (TAM)	FF {K}	[FLG.1] fF1	Easy access to Flag K (111)
Flag L	Flag L	EasyAccess (TAM)	FF {L}	[FLG.1] fF2	Easy access to Flag L (108)
Flag M	Flag M	EasyAccess (TAM)	FF {M}	[FLG.1] F1	Easy access to Flag M (211)
FP	Fractional Part	Function (monadic)	#TAM [LN] (F) for closed number	[REAL.1] F2	Fractional Part of real number
G	Local label G	EasyAccess (TAM)	LBL {G}		Easy access to local label G
GT0.	Go to label or step	Command (nonpgm)	GT0 - [.] or [XEQ] - [.] to enter GT0.		GT0. to go to (local) label or step ; GT0.. to move the program pointer to the end of program memory
GT0↑	Go to program start	Command (nonpgm)	GT0 (g[XEQ]) - [+]		GT0↑ to go to the start of the (previous) program or jump to the start of program memory
GT0↓	Go to program end	Command (nonpgm)	GT0 (g[XEQ]) - [+]		GT0↓ to go to the next program or jump to the end of program memory
H	Local label H	EasyAccess (TAM)	LBL {H}		Easy access to local label H
HEX	Hexadecimal	Setting (pgm)	#TAM [RCL] (H)	[BASE.1] F1	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
HOME	HOME	MENU	longpress [<f>] (because KEY.LP is SET; also available in Normal Mode of PEM; it is indicated by the white keyboard label «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
I	Local label I	EasyAccess (TAM)	LBL {I}		Easy access to local label I
i	Complex i	Character	alpha g[x ²]	[αMath.3] gF6	Character i
IP	Integer Part	Function (monadic)	#TAM [R+] (I) for closed number	[REAL.1] F1	Integer Part of number (returns Real number)
J	Local label J	EasyAccess (TAM)	LBL {J}		Easy access to local label J
j	Complex j	Character	alpha g[DRG]	[αMath.4] F1	Character j
K	Local label K	EasyAccess (TAM)	LBL {K}		Easy access to local label K
L	Local label L	EasyAccess (TAM)	LBL {L}		Easy access to local label L

Label	Fullname	Type	Combo	Regular location	Description
LAP→R	Store lap time in register	Command	[.] (STOPW active)	[STOPW.1] F4	Store lap time in register, increment CRA (Current Register Address) and reset lap timer ; subscript shows CRA
LAP→Σ	Add lap time to STATS	Command	[+] (STOPW active)	[STOPW.1] F2	Add lap time to STATS (statistics matrix) and reset lap timer ; subscript shows number of data points
MSQRT	Square root of matrix	Function (monadic)	[√x]	[MATX.2] F2	Square root of matrix in X : if $M = \text{MSQRT}(X)$, $M \times M = X$, where $\times$ denotes matrix multiplication
MyMenu	MyMenu	MENU	longpress [<g>] (because KEY.LP is SET; also available in Normal Mode of PEM; it is hinted at by the white keyboard label «CUST»); during assignment, pressing USER (f[DRG]) opens 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.R47
Myα	MyAlpha	MENU	alpha long [EXIT] ; EIM long [EXIT] ; alpha long [<g>] (because KEY.LP is SET; also available in alpha mode of PEM)	[ALPHA.1] fF1	User menu to quickly access user selected special characters ; all buttons are user assignable ('riveted' look) ; initially unpopulated ; reset using Myα.R
NUM	Numlock	Setting	alpha f[+] / alpha f[+]	[ALPHA.1] gF2	Switch on numlock ; not programmable - in PEM, use flag NUM
P.FN	Programming functions	MENU	PEM long [EXIT]	[P.FN3.1] F6 ; g[R/S]	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
R/S	Start/stop stopwatch	Command	[R/S] (STOPW active)	[STOPW.1] F5	Start/stop the stopwatch
RCL	Recall and load to timer	Command (nonpgm)	[RCL] (STOPW active)	[STOPW.1] fF3	Recall register value and load to stopwatch timer (decimal hours or time value)
RCL-	Recall and subtract	Function (monadic)	[RCL] - [-]		Recall register or variable and subtract X
RCL↑	Recall maximum	Command	[RCL] - [+]		Recall maximum of X and register or variable
RCL↓	Recall minimum	Command	[RCL] - [+]		Recall minimum of X and register or variable
RCL+	Recall and add	Function (monadic)	[RCL] - [+]		Recall register or variable and add X
RCL÷	Recall and divide	Function (monadic)	[RCL] - [÷]		Recall register or variable and divide by X
RCL×	Recall and multiply	Function (monadic)	[RCL] - [×]		Recall register or variable and multiply by X
RCLCFG	Recall config	Command	[RCL] - Config		Recall config from register or variable
RCELE	Recall matrix element	Command	[RCL] - ...EL	[MATX.3] gF4	Recall current element of indexed matrix
RCLIJ	Recall current matrix index	Command	[RCL] - ...IJ	[MATX.3] F4	Recall current matrix index (row I and column J)
RCLS	Recall stack	Command	[RCL] - Stack		Recall complete stack from 4 or 8 registers
RCLVEL	Recall vector element	Command	[RCL] - ...VEL _{nn}	[MATX.1] gF4	Recall element n (TAM) of vector or matrix in X
RegA	Register A	EasyAccess (TAM)	STO/RCL {A}	[REG.1] fF3 ; TAM [x ²]	Easy access to Register A (104) : Stack level 5
RegB	Register B	EasyAccess (TAM)	STO/RCL {B}	[REG.1] fF4 ; TAM [√x]	Easy access to Register B (105) : Stack level 6
RegC	Register C	EasyAccess (TAM)	STO/RCL {C}	[REG.1] fF5 ; TAM [1/x]	Easy access to Register C (106) : Stack level 7
RegD	Register D	EasyAccess (TAM)	STO/RCL {D}	[REG.1] fF6 ; TAM [y ^x]	Easy access to Register D (107) : Stack level 8
RegE	Register E	EasyAccess (TAM)	STO/RCL {E}	[REG.2] F1 ; TAM [LOG]	Easy access to Register E (217) : General purpose register E
RegF	Register F	EasyAccess (TAM)	STO/RCL {F}	[REG.2] F2 ; TAM [LN]	Easy access to Register F (218) : General purpose register F
RegG	Register G	EasyAccess (TAM)	STO/RCL {G}	[REG.2] F3 ; TAM [STO]	Easy access to Register G (219) : General purpose register G ; also used to store third to last long integer prime factor for FACTORS
RegH	Register H	EasyAccess (TAM)	STO/RCL {H}	[REG.2] F4 ; TAM [RCL]	Easy access to Register H (220) : General purpose register H ; also used to store second to last long integer prime factor for FACTORS
RegI	Register I	EasyAccess (TAM)	STO/RCL {I}	[REG.1] gF1 ; TAM [R4]	Easy access to Register I (109) : General purpose register I ; also used to store current row of indexed matrix
RegJ	Register J	EasyAccess (TAM)	STO/RCL {J}	[REG.1] gF2 ; TAM [DRG]	Easy access to Register J (110) : General purpose register J ; also used to store current column of indexed matrix
RegK	Register K	EasyAccess (TAM)	STO/RCL {K}	[REG.1] fF1 ; TAM [x2y]	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
RegL	Register L	EasyAccess (TAM)	STO/RCL {L}	[REG.1] fF2 ; TAM [CHS]	Easy access to Register L (108) : Last X
RegM	Register M	EasyAccess (TAM)	STO/RCL {M}	[REG.1] F1 ; TAM [EEX]	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
REGS	Register browser	Browser	[.] : Switch register/variable view, [R/S] : Switch contents/storage view, [RCL] : Recall bottom item, [+], [+], , 00..99 : Navigate, [EXIT] : Exit Register browser	[STOPW.1] fF2 ; g[+]	Browse all registers
RESET	Reset stopwatch	Command	[+] (STOPW active)	[STOPW.1] F6	Reset the stopwatch
π	Subscript pi	Character	alpha g[+]π (alpha g[R+])		Character π

Label	Fullname	Type	Combo	Regular location	Description
SNAP	Screenshot	Command	[<f>] + [CHS] (DMCP-shortcut for screenshot bitmap only); longpress [EXIT]; NIM long [EXIT]; PEM longer [EXIT]	[I/O.1] gF5 ; alpha g [EXIT]	Save screenshot to backup file (.bmp) and save contents of stack or alpha buffer as backup file (.REGS.TSV) in FAT ; plays clicking sound
SST	Single step	Command (nonpgm)	[↵]	f[↵]	Single step
Stack	Store stack	Command (TAM)	[STO] - Stack	[TamSto.1] fF2	Presented in menu TamSto to store entire stack in 4 or 8 registers
Stack	Recall stack	Command (TAM)	[RCL] - Stack	[TamRcl.1] fF2	Presented in menu TamRcl to recall complete stack from 4 or 8 registers
STO-	Subtract X from register	Function (monadic)	[STO] - [-]		Subtract X from register or variable
STO↑	Store maximum	Command	[STO] - [↑]		Store maximum of X and register or variable
STO↓	Store minimum	Command	[STO] - [↓]		Store minimum of X and register or variable
STO+	Add X into register	Function (monadic)	[STO] - [+]		Add X into register or variable ; if X is a text string, the target is converted to text string and X is concatenated
STO÷	Divide X into register	Function (monadic)	[STO] - [÷]		Divide register or variable by X
STO×	Multiply X into register	Function (monadic)	[STO] - [×]		Multiply register or variable by X
STOCFG	Store config	Command	[STO] - Config		Store config in register or variable
STOEL	Store X into matrix element	Command	[STO] - Config	[MATX.3] gF3	Store X into current element of indexed matrix
STOIJ	Set current matrix index	Command	[STO] - ...IJ	[MATX.3] F3	Set current matrix index (row I and column J)
STOPW	Stopwatch	MENU (internal)	[0]... : Direct entry of digit 0... into CRA (Current Register Address), ...[9] : Direct entry of digit ...9 into CRA (Current Register Address), [↑] : Increment CRA (Current Register Address), [↓] : Decrement CRA (Current Register Address), alpha g [EXIT] : Save screenshot to backup file (.bmp) and save contents of stack or alpha buffer as backup file (.REGS.TSV) in FAT (SNAP), [EXIT] : Exit the stopwatch		Stopwatch with running time, counter and countdown (playing BEEP after running down)
STOS	Store stack	Command	[STO] - Stack		Store entire stack in 4 or 8 registers
STOVEL	Store X into vector element	Command	[STO] - ...VEL _{nn}	[MATX.1] gF3	Store X into element n (TAM) of vector or matrix in Y
^{SUS}	Normal-super-subscript	Setting (cyclic)	alpha g [↑] / alpha g [↓]	[ALPHA.1] gF3	Cycle normal-super-subscript (locking)
TamAlpha	TAM mode (alpha)	MENU (TAM)	Activated from TAM menus by α (f[XEQ]) or TAM [XEQ]	[Tam.1] f[XEQ]	Transient Alpha Mode is activated for trailing input (alpha)
TamIndirect	TAM mode (indirect)	MENU (TAM)	Activated from TAM menus by Indirection (→)	[Tam.1] F1 (Indirection)	Transient Alpha Mode is activated for trailing input (indirect)
TamLocalLabel	TAM mode (local label)	MENU (TAM)	Activated from TAM menus by Colon button (:))		Transient Alpha Mode is activated for trailing input (local label)
TIM→R	Store time in register	Command	[ENTER] (STOPW active)	[STOPW.1] F3	Store time in register and increment CRA (Current Register Address) ; subscript shows R/S
TIM→Σ	Add time to STATS	Command	[Σ+] (STOPW active)	[STOPW.1] F1	Add time to STATS (statistics matrix) ; subscript shows number of data points
X.EDIT	Edit X	Command	alpha long [ENTER] ; EIM long [ENTER]	[ALPHA.1] fF2 ; alpha f [ENTER]	Edit contents of stack register X as text
α	Alpha input	MENU (item)	longpress [XEQ] (even when M.14 is ON)	f[XEQ]	Open alpha menu to enter, edit and clear alpha input ; when TAM is active, open menu TamAlpha
α	alpha	Character	alpha f [XEQ]	[α...ω.1] F1	Character α
Δ	DELTA	Character	alpha g [RCL]	[A...Ω.1] F4	Character Δ
π	pi	Character	alpha g [R↓]	[α...ω.1] gF5	Greek character π