

R47v0 Keyboard row 2									
Key	Row	Column	Kind	Label	FullName	Extended description	Type	Catalog	Default
R47v0.21.11	2	1		$x^2$	Square	Square of X	Function (monadic)	$x^2$	
R47v0.21.12	2	1	f	$i$	Complex number (rectangular)	Enter complex number (rectangular) 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 CPXj when in RECT mode)	Command	op_ $i$	
R47v0.21.13	2	1	g	$\rightarrow R$	To rectangular	Transform polar to rectangular coordinates (stack conventions according to flag RP <sub>HP</sub> ) ; transform complex number to rectangular notation (monadic) and set RECT tag (TI : x : Re = ; y : Im = (2 stack levels))	Function (monadic ; dyadic)	$\rightarrow$ RECT	
R47v0.21.31	2	1	alpha	A	A	Character A (Code : 65)	Character		
R47v0.21.32	2	1	alpha f	a	a lowercase	Character a (Code : 97)	Character		
R47v0.21.33	2	1	alpha g	$i$	op_i	Character $i$ (Hidden : alpha g [ $x^2$ ] ; Code : 8520)	Character		
R47v0.22.11	2	2		$\sqrt{x}$	Square root	Square root	Function (monadic)	$\sqrt{x}$	
R47v0.22.12	2	2	f	$i_{\theta}$	Complex number (polar)	Enter complex number (polar) whether POLAR is set or not ; e.g. a $i_{\theta}$ b ENTER results in a $r$ b (according to ADM) (Info : In NIM, works like CC with POLAR set ; displayed according to flag CPXj when in RECT mode)	Command	op_ $i_{\theta}$	
R47v0.22.13	2	2	g	$\rightarrow P$	To polar	Transform rectangular to polar coordinates (stack conventions according to flag RP <sub>HP</sub> or ADM tag) ; transform complex number to polar notation (monadic) and set POLAR tag (TI : r = ; $\theta$ = (2 stack levels))	Function (monadic ; dyadic)	$\rightarrow$ POLAR	
R47v0.22.31	2	2	alpha	B	B	Character B (Code : 66)	Character		
R47v0.22.32	2	2	alpha f	b	b lowercase	Character b (Code : 98)	Character		
R47v0.22.33	2	2	alpha g	$\sqrt{\phantom{x}}$	Square root	Character $\sqrt{\phantom{x}}$ (Hidden : alpha g [ $\sqrt{x}$ ] ; Code : 8730)	Character		
R47v0.23.11	2	3		$1/x$	Reciprocal	Reciprocal (1/x) (Info : When X is a matrix $1/x$ inverts it ( $[M]^{-1}$ ))	Function (monadic)	$1/x$	
R47v0.23.12	2	3	f	x!	Factorial x! ; $\Gamma(x+1)$	For integers : x! ; for reals : $\Gamma(x+1)$ (Info : Max integer: 450 ; max real : 2123.549 956 662 463 236 31 ; integers > max are converted to reals)	Function (monadic)	x!	
R47v0.23.13	2	3	g	.ms	Minutes & seconds	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)	Function (cyclic ; monadic)	.ms	
R47v0.23.31	2	3	alpha	C	C	Character C (Code : 67)	Character		
R47v0.23.32	2	3	alpha f	c	c lowercase	Character c (Code : 99)	Character		
R47v0.23.33	2	3	alpha g	!	Exclamation mark	Character ! (Hidden : alpha g [ $1/x$ ] ; Code : 33)	Character		
R47v0.24.11	2	4		$y^x$	y to the power x	Raise value in the Y-register to the power in the X-register	Function (dyadic)	$y^x$	
R47v0.24.12	2	4	f	$\sqrt[x]{y}$	xth root	Xth root of Y	Function (dyadic)	$\sqrt[x]{y}$	
R47v0.24.13	2	4	g	.d	Decimal	Convert to decimal (real) value ; clear fraction mode, base mode ; convert degrees / hours / date to real ; convert NIM input to date (according to date format set and implied conversion set by YY) ; convert complex number with zero imaginary part to real number ; in Program Entry Mode $\rightarrow$ REAL is entered (TI (degrees ; hours ; date) : decimal <sup>o</sup> ; decimal h ; yyyy-mm-dd: )	Function (monadic)		
R47v0.24.31	2	4	alpha	D	D	Character D (Code : 68)	Character		
R47v0.24.32	2	4	alpha f	d	d lowercase	Character d (Code : 100)	Character		
R47v0.24.33	2	4	alpha g	^	Circumflex accent	Character ^ (Hidden : alpha g [ $y^x$ ] ; Code : 94)	Character		
R47v0.25.11	2	5		LOG	Common logarithm	Common logarithm (base 10)	Function (monadic)	LOG	
R47v0.25.12	2	5	f	$10^x$	10 to the power x	Raise 10 to the power in the X-register	Function (monadic)	$10^x$	
R47v0.25.13	2	5	g	$\rightarrow I$	To integer	Convert to long integer/short integer (cyclic, max 1000 digits) (Info : Shortint indicated by subscript <sub>10</sub> ; can show TI: 0vrfl<0: or 0vrfl>64bits: which can be abbreviated as OF, indicating overflow condition)	Function (cyclic ; monadic)	$\rightarrow$ I	
R47v0.25.31	2	5	alpha	E	E	Character E (Code : 69)	Character		
R47v0.25.32	2	5	alpha f	e	e lowercase	Character e (Code : 101)	Character		
R47v0.25.33	2	5	alpha g	$\varnothing$	Euler's e	Character $\varnothing$ (Hidden : alpha g [LOG] ; Code : 8519)	Character		
R47v0.26.11	2	6		LN	Natural logarithm	Natural logarithm (base e)	Function (monadic)	LN	
R47v0.26.12	2	6	f	$e^x$	e to the power x	Raise e to the power in the X-register	Function (monadic)	$e^x$	
R47v0.26.13	2	6	g	#	Number base	Set number base ; operates on all stack registers depending on BASE <sub>HP</sub> ; reset by [.d] (g [ $y^x$ ]) (TAM : $\rightarrow$ INT __ TamNonReg menu ; #TAM shortcuts : B = BIN ; D = DEC ; ENTER = DEC ; H = HEX ; Info : SBI depends on SBfrac)	Setting (pgm ; stack)	$\rightarrow$ INT	
R47v0.26.31	2	6	alpha	F	F	Character F (Code : 70)	Character		
R47v0.26.32	2	6	alpha f	f	f lowercase	Character f (Code : 102)	Character		
R47v0.26.33	2	6	alpha g	#	Number sign	Character # (Hidden : alpha g [LN] ; Code : 35)	Character		