GNU Calc Reference Card
(for GNU Emacs version 26)

Starting and Stopping

start/stop standard Calc C-x * c
start/stop X keypad Calc C-x * k
start/stop either: C-x * *
stop standard Calc q
Calc tutorial t
run Calc in other window o
quick calculation in minibuffer q

Getting Help

The h prefix key is Calc's analogue of C-h in Emacs.

quick summary of keys ?
describe key briefly h c
describe key fully h k
describe function or command h f
read Info manual i or C-x * i
read full Calc summary h s or C-x * s

Error Recovery

abort command in progress C-g
display recent error messages w
undo last operation U
redo last operation D
recall last arguments M-RET
edit top of stack : n
reset Calc to initial state C-x * 0 (zero)

Transferring Data

grab region from a buffer C-x * g
grab rectangle from a buffer C-x * r
grab rectangle, summing columns C-x * :
grab rectangle, summing rows C-x * -
yank data to a buffer C-x * y
Also, try C-k/C-y or X cut and paste.

Examples

In RPN, enter numbers first, separated by RET if necessary, then type the operator. To enter a calculation in algebraic form, press the apostrophe first.

<table>
<thead>
<tr>
<th>RPN style:</th>
<th>algebraic style:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Example:</td>
</tr>
<tr>
<td>2 RET 3 +</td>
<td>' 2+3 RET</td>
</tr>
<tr>
<td>Example:</td>
<td>2 RET 3 + 4 *</td>
</tr>
<tr>
<td>2 RET 3 + 4</td>
<td>' (2+3)*4 RET</td>
</tr>
<tr>
<td>Example:</td>
<td>2 RET 3 RET 4 +</td>
</tr>
<tr>
<td>2*(3+4) RET</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>3 RET 6 + Q 3 -</td>
</tr>
<tr>
<td>sqrt(3*6)-3 RET</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>P 3 / n S</td>
</tr>
<tr>
<td>' sin(-pi/3) RET</td>
<td></td>
</tr>
</tbody>
</table>

Arithmetic

add, subtract, multiply, divide +, -, *, /
raise to a power, nth root ",", "^"
change sign",
reciprocal 1/x k
square root \(\sqrt{ }\) Q
set precision p
c 2
convert to fraction, float c F, c f
enter using algebraic notation 2+3*4
refer to previous result 3^2*2
refer to higher stack entries $1+$2^2
finish alg entry without evaluating LFD
set mode where alg entry used by default m a

Stack Commands

Here \(S_n\) is the \(n\)th stack entry, and \(N\) is the size of the stack.

key no prefix prefix n \(\rightarrow\) n
RET copy \(S_1\) copy \(S_1\) = copy \(S_n\)
LFD copy \(S_2\) copy \(S_2\) = copy \(S_{1..n}\)
DEL delete \(S_1\) delete \(S_1\) = delete \(S_n\)
M-DEL delete \(S_2\) delete \(S_2\) = delete \(S_{1..n}\)
TAB swap \(S_1\leftrightarrow S_2\) roll \(S_1\) to \(S_n\) roll \(S_n\) to \(S_{1..n}\)
M-TAB roll \(S_1\) to \(S_n\) roll \(S_n\) to \(S_1\) With a 0 prefix, these copy, delete, or reverse the entire stack.

Display

scroll horizontally, vertically \(<\), \(>\), \{\} home cursor o
line numbers on/off d l trail display on/off t d scientific notation d s fixed-point notation d f
floating-point (normal) notation d n group digits with commas d g
For display mode commands, h prefix prevents screen redraw and I prefix temporarily redraws top of stack.

Notations

scientific notation 6.02e23
minus sign in numeric entry _23 or _23 n
fractions 3:4
(s, y) (r; \theta)
complex numbers [1, 2, 3]
matrices (or nested vectors) [1, 2; 3, 4]
error forms (p key) 100 \(\pm\) 0.5
interval forms \(\{2 \ldots 5\}\)
modulo forms (M key) 6 mod 24
HMS forms 50 30' 0"
date forms \(<Jul 4, 1992>\)

Scientific Functions

ln, log_{10}, log_b exponential \(e^x\), \(10^x\) \(\sin\), cos, tan \(\arcsin\), \(\arccos\), \(\arctan\)
inverse, hyperbolic prefix keys \(I\)
two-argument arctan \(f T\)
degrees, radians modes \(m d\), \(m r\)
pi \(\pi\)
factorial, double factorial \(!\), \(\frac{\pi}{2}\)
combinations, permutations \(k c\), \(H k c\)
prime factorization \(k f\)
next prime, previous prime \(k n\), \(k I n\)
GCD, LCM \(k g\), \(k l\)
random number, shuffle \(k r\), \(k h\)
minimum, maximum \(f n\), \(f x\)
error functions erf, erfc \(f e\), \(f f e\)
gamma, beta functions \(f g\), \(f B\)
incorporate gamma, beta functions \(f G\), \(f B\)
Bessel \(J_n\), \(Y_n\) functions \(f j\), \(f y\)
complex magnitude, arg, conjugate \(A\), \(G\), \(J\)
real, imaginary parts \(f r\), \(f i\)
convert polar/rectangular \(c p\)

Financial Functions

enter percentage \(M\%\)
convert to percentage \(c\%\)
percentage change \(b\%\)
present value \(b P\)
future value \(b F\)
rate of return \(b T\)
number of payments \(b N\)
size of payments \(b M\)
net present value, int. rate of return \(b H\), \(b I\)
Above computations assume payments at end of period. Use I prefix for beginning of period, or \(H\) for a lump sum investment.
straight-line depreciation \(b S\)
sum-of-years'-digits \(b V\)
double declining balance \(b D\)

Units

enter with units \(7\) \(55\) mi/hr
convert to new units, base units \(u c\), \(u b\)
convert temperature units \(u t\)
simplify units expression \(u s\)
view units table \(u v\)

Common units:
distance: \(m\), \(cm\), \(mm\), \(km\); \(in\), \(ft\), \(mi\), \(mfd\); point, \(lyr\)
volume: \(l\) or \(m\); \(gal\), \(qt\), \(pt\), \(cup\), \(floz\), \(tbsp\), \(tsp\)
mass: \(g\), \(mg\), \(kg\), \(t\); \(lb\), \(oz\), \(ton\)
time: \(s\) or \(sec\), \(ms\), \(us\), \(ns\), \(min\), \(hr\), \(day\), \(wk\)
temperature: \(degC\), \(degF\), \(K\)

© 2018 Free Software Foundation, Inc. Permissions on back.
**GNU Calc Reference Card**

**Programmer’s Functions**

- Binary, octal, hex display: `d 2, d 8, d 6`
- Decimal, other radix display: `d 0, d r`
- Display leading zeros: `d z`
- Entering non-decimal numbers: `16#7FFF`
- Binary word size: `b w`
- Binary AND, OR, XOR: `b a, b o, b x`
- Binary DIFF, NOT: `b d, b n`
- Left shift: `b l`
- Logical right shift: `b r`
- Arithmetic right shift: `b R`
- Integer quotient, remainder: `\ b n`, `\ d N`
- Integer square root, logarithm: `f I F`, `f Q`, `f I`
- Logical right shift: `b r` (left shift: `b l`)
- Binary DIFF, NOT: `b d` (binary: `b w`)
- Entering non-decimal numbers: `16#7FFF`
- Display leading zeros: `d r`
- Binary, octal, hex display: `d X` (decimal: `d r`)

**Variables**

- Variable names are single digits or whole words.
- Store to variable: `s t`
- Store and keep on stack: `s s`
- Recall from variable: `s r`
- Shorthand for digit variables: `t n, s n, r n`
- Unstore, exchange variable: `s u, s x`
- Edit variable: `e a`

**Vector Operations**

- Vector of 1, 2, …, n: `v x n`
- Vector of n counts from a by b: `C-u v x`
- Vector of copies of a value: `v b`
- Concatenate into vector: `v |`
- Pack many stack items into vector: `v p`
- Unpack vector or object: `v u`
- Length of vector (list): `v l`
- Reverse vector: `v v`
- Sort, grade vector: `v S`, `v G`
- Histogram of vector data: `V H`
- Extract vector element: `v x`
- Matrix determinant, inverse: `V D`, `v t`, `v T`
- Cross product: `V C`, `v *`
- Identity matrix: `v i`
- Extract matrix row, column: `v r, v c`
- Intersect, union, diff of sets: `v ^`, `v -`, `V V`, `V -`
- Cardinality of set: `v #`
- Add vectors elementwise (i.e., `map +`): `V M +`
- Sum elements in vector (i.e., `reduce +`): `V R +`
- Sum columns in matrix: `V R - +`
- Sum elements, accumulate results: `V U +`

**Algebra**

- Enter an algebraic formula: `' 2x + 3y^2`
- Enter an equation: `' 2x = 2y + 18`
- Symbolic (vs. numeric) mode: `m s`
- Fractions (vs. float) mode: `m f`
- Suppress evaluation of formulas: `m O`
- Return to default evaluation rules: `m D`
- “Big” display mode: `d B`
- C, Pascal, FORTRAN modes: `d C, d P, d F`
- Q, X, LISP, eqn modes: `d T, d L, d E`
- Maxima: `d X`
- Unformatted mode: `d U`
- Normal language mode: `d N`
- Simplify formula: `a s`
- Put formula into rational form: `a n`
- Evaluate variables in formula: `a n`
- Evaluate numerically: `a n`
- Let variable equal a value in formula: `s 1 x = val`
- Declare properties of variable: `s d`
- Common decks: `pos`, `int`, `real`, `scalar`, `[a..b]`
- Expand, collect terms: `a x`, `a c`
- Factor, partial fractions: `a a`, `a a`
- Polynomial quotient, remainder, GCD: `a \ a`, `a a`, `a g`
- Derivative, integral: `a d`, `a i`
- Taylor series: `a t`
- Principal solution to equation(s): `a S`
- List of solutions: `a P`
- Generic solution: `a H S`
- Apply function to both sides of eqn: `a M`
- Rewrite formula: `a r`
- Example: `a r a + b + a * c := a * (b + c)`
- Example: `a r sin(x)^2 := 1 - cos(x)^2`
- Example: `a r cos(n*p + 1) := : integer(n) ~/ n%2 == 0`
- Example: `a r [(f(x) := : n) * (f(x - 1) := : n > 0)]`
- Put rules in `AlgSimpRules` to apply during `a a` command.
- Common markers: `opt`, `plain`, `quote`, `eval`, `let`, `remember`.

**Numerical Computations**

- Sum formula over a range: `a +`
- Product of formula over a range: `a *`
- Tabulate formula over a range: `a T`
- Integrate numerically over a range: `a I`
- Find zero of formula or equation: `a R`
- Find local min, max of formula: `a N`, `a X`
- Fit data to line or curve: `a F`
- Mean of data in vector or variable: `u M`
- Median of data: `u M`
- Geometric mean of data: `u G`
- Sum, product of data: `u +`, `u *`
- Minimum, maximum of data: `u N`, `u X`
- Sample, population standard deviation: `u S`, `u I S`

**Selections**

- Select subformula under cursor: `j a`
- Select nth subformula: `j n`
- Select more: `j a`
- Unselect this, all formulas: `j u`, `j c`
- Copy indicated subformula: `j R E T`
- Delete indicated subformula: `j D E L`
- Commute selected terms: `j C`
- Commute term leftward, rightward: `j l`, `j r`
- Distribute, merge selection: `j D`, `j M`
- Isolate selected term in equation: `j I`
- Negate, invert term in context: `j M`, `j &`
- Rewrite selected term: `j r`

**Graphics**

- Graph function or data: `g f`
- Graph 3D function or data: `g F`
- Replot current graph: `g p`
- Print current graph: `g a`
- Add curve to graph: `g a`
- Set number of data points: `g M`
- Set line, point styles: `g S`, `g S`
- Set log vs. linear: `g L`, `g L`
- Set range for x, y axis: `g R`, `g R`
- Close graphics window: `g q`

**Programming**

- Begin, end recording a macro: `C-x * m`
- Replace keyboard macro: `C-x * m`
- Read region as written-out macro: `Z`, `Z`, `Z`
- If, else, end if: `Z`, `Z`, `Z`
- Equal to, less than, member of: `a =`, `a <`, `a`, `{`
- Repeat n times, break from loop: `Z`, `Z`, `Z`, `/`
- “For” loop: start, end; body, step: `Z`, `Z`, `Z`
- Save, restore mode settings: `Z`, `Z`, `Z`
- Query user during macro: `Z`
- Put finished macro on a key: `Z`
- Define function with formula: `Z F`
- Edit definition: `Z E`
- Record user-defined command permanently: `Z P`
- Record variable value permanently: `Z P`
- Record mode settings permanently: `Z m`

---

Copyright © 2018 Free Software Foundation, Inc. Designed by Dave Gillespie and Stephen Gildea, for GNU Emacs Calc. Released under the terms of the GNU General Public License version 3 or later. For more Emacs documentation, and the TeX source for this card, see the Emacs distribution, or https://www.gnu.org/software/emacs