

Table of contents

Introduction

[The Mighty Midget](#)

[Midget as shareware](#)

[How to register](#)

[How to reach me](#)

[Copyrights and credits](#)

[Meet Midget's Big Mama](#)

Reference

[Constants](#)

[Functions](#)

[Operators](#)

[Variables](#)

[Expression syntax](#)

[Implied multiplication](#)

[Angular mode](#)

[Result format](#)

[Accuracy](#)

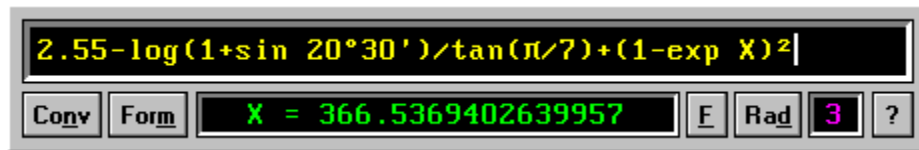
[Unit conversion](#)

[The pop-up menu](#)

[Enhanced features](#)

Welcome to Midget, the calculator for the rest of us!

I hope you like the program and find it useful. Note that Midget is shareware: check out [what it means](#) and [how to register](#).



For starters, here is the Midget window. Click on various objects in the picture to get a brief explanation on what they are used for.

Using Midget is a no-brainer. To calculate the value of an expression, e.g. "2.5+3/7-log 13", just type it in and hit the Enter key. This is it.

Instead of pressing Enter, you can use the "=" key; the result will then be stored in a variable (i.e. abbreviated as an uppercase letter), chosen from a pop-up menu.

In order to move Midget around the screen, click and hold the left mouse button at the window edge (the cursor will change) and drag the Midget window around.

For more, look into:

[Table of contents](#)

Midget as shareware

Midget is distributed as shareware: you are granted a trial license for a period of 30 days to see whether you will find the program useful and worth the (very modest) price I'm asking.

After that you have the options of either registering (purchasing) Midget, or removing it from your system. Not doing so puts you in violation of U.S. and international copyright laws, but, first of all, it is just unethical.

If you decide to keep Midget, registering (i.e. buying it) will bring you additional benefits: unlocking of the enhanced features, free upgrades, new version notifications and technical support via email. Last but not least, the program will also get rid of the nagging screen which will start showing up after the trial license expires.

See also:

[How to register](#)

[How to reach me](#)

How to register

Simple: send a check or a money order for U.S. \$10 (this you can afford!) to my address:

Dr. J. Andrzej Wrotniak
2057 Happy Lane
Crofton, MD 21114-1917
USA

You do not have to include any letter: just state "Midget" in the check memo space. (Of course, letters and comments are welcome.)

If you have an email address, include it with the order (or just write on the check): I will email you promptly a registration file, MIDGET.KEY.

If you do not have an email address, I will ship you a 3.5" floppy with the program (state which Windows you are using; otherwise I will have to include both). There are no extra charges for this, regardless of where you live.

Sorry, I cannot afford to accept credit card orders. The registration price is so low that the handling fees would eat 60% of it, and I want very much to keep that price down.

Registering on the CompuServe

The CompuServe subscribers may register online: GO SWREG and search for Midget or Wrotniak. The registration fee of \$10 will be added to your next CompuServe bill, and you should receive the registration key (usually) within less than 24 hours.

See also:

[How to reach me online](#)

How to reach me

The fastest, most convenient and cheapest way is via email to my CompuServe address:

70611,2552 (J.Andrzej Wrotniak)

If you are not on the CompuServe, use the Internet version:

70611.2552@compuserve.com

(please note the dot instead of a comma).

You may also send me email while visiting my personal Web page at

<http://www.freeflight.com/wrotniak/>

where new versions of all my programs are posted usually before they are available anywhere else.

If you do not have an access to email, use the good old U.S. Mail.

Copyrights and credits

Midget is copyright © 1996 by J. Andrzej Wrotniak. All rights reserved.

This software has been developed using Borland's Delphi and contains parts of code copyright © 1995-96 by Borland International.

The on-line help has been written with the Visual Help from WinWare, Inc.

The sound files supplied with Midget have been created by Ted Tatman (thanks!) and are in public domain.

Midget inherited many design ideas and a significant amount of code from two of my previous projects:

- from Ex, a Scientific Calculator for the HP LX Palmtop computers (1992-95) -- expression parser, most of the underlying math engine
- from Kalkulator, the Mother of All Calculators (1995-96) -- basic parts of the user interface, most of the Windows code

[Important: read the enclosed disclaimer before using the program!](#)

Legal Disclaimer

Midget is licensed on the "as is" basis.

Although the author has extensively tested the software and reviewed the documentation, he makes no warrant or representation, explicit or implied, with respect to this package, its quality or performance, merchantability or fitness for any particular purpose, in any hardware or software environment.

In no event shall the author be liable for direct, indirect, special, incidental or consequential damages, arising from use, misuse, or inability to use the software or documentation.

The Mother of All Calculators



If your numerical needs exceed the capabilities of Midget, you should meet Midget's Big Mama: the Kalkulator. (Click on the icon to see a screen shot.)

Kalkulator is probably the most powerful program of its kind on the market. In addition to enhancing the capabilities existing in Midget, it offers:

- A wider function set: hyperbolic, Euler, statistical distributions (including integral and inverse-integral)
- Pre-defined physical constants
- Numerical operations on expressions: integration, computing derivatives, finding zeros of functions, indexed sums
- Function plotting (with smart handling of line breaks and changes in curvature)
- Statistical operations on points in a data buffer: statistical parameters, histograms, polynomial regression (up to the ninth degree), column-wise arithmetics on data
- Finding roots of polynomial equations
- Linear algebra: vector and matrix arithmetics, matrix inversion, systems of linear equations

The June'96 release will include a binary, octal and hex calculator; function optimization and differential equations are expected in December.

You can get a functional evaluation copy of Kalkulator from the CompuServe (the Science Forum, Math Library) or from the [World Wide Web](#).

Constants

Numeric constants

Constants are used as components of expressions. A constant consists of digits 0..9, an optional decimal point and an optional decimal exponent symbol, "E"; the exponent may be signed or not. For example, "1.5E-3" means the same as "0.0015". A zero preceding the decimal point can be skipped, as in ".0015".

A leading minus (or plus) sign is not treated as a part of a constant, but rather as an operator, although this is a purely academic distinction.

When an expression is reformatted before being computed, the constants in it may become reformatted as well, to a (hopefully) simpler representation.

Important for European users: regardless of your regional settings, Kalkulator always uses a point, ".", as the decimal separator.

Angular constants

These are entered with use of digits and the symbols of degrees, minutes and seconds. To enter the degree symbol from the keyboard, use Ctrl-d; minutes are entered as a single quote, and seconds -- as a double quote.

Symbolic constants

There are two predefined symbolic constants: pi (3.14...) and Euler's e (2.71...). They are denoted with symbols π and e , respectively. To enter them from the keyboard, use the Ctrl-p and Ctrl-e keystroke combinations.

Functions

Functions are used as elements of expressions in a customary way, e.g. "log(.25+X)-sqr(7/8)". Their names are spelled in lower case.

Most functions accept one argument. If this argument is a constant or a variable, it does not have to be enclosed in parentheses. Multiple arguments are separated with commas, like in "rad(1,C+2.5)".

The following functions are available in Midget:

Roots and powers: [sqr](#), [sq](#), [cur](#), [cu](#)

Logarithmic and exponential: [ln](#), [log](#), [exp](#), [ten](#)

Trigonometric: [sin](#), [cos](#), [tan](#), [cot](#)

Inverse trigonometric: [asin](#), [acos](#), [atan](#), [acot](#)

Polar coordinate functions: [rad](#), [ang](#)

Unit conversion: [lbs](#), [ft](#), [hrs](#)

Others: [abs](#), [int](#), [fra](#), [sgn](#), [min](#), [max](#)

See also:

[Operators](#)

Operators

The arithmetic operators available in Midget include:

Basic four operations: +, -, *, /

Raising to power: ^, like in "X^1.5"

Square and cube: ², ³

Factorial: !

Implied multiplication: · (a tiny dot, sometimes not shown)

When entered from the keyboard, some of the operators will require special keystroke combinations.

The multiplication operator, shown in the display as "×", is entered with the asterisk key, "**", which is a commonly accepted procedure. (In the examples shown here we will be using the asterisk as well.)

See also:

[Expression syntax](#)

Expression syntax

The expression syntax follows closely the customary "blackboard" notation. The maximum expression length is 44 characters or 36 tokens (operators, variables, constants or separators); in the registered version these limits have been increased to 120 characters or 72 tokens.

The following is a formal description of the syntax; if it seems too technical, just skip it and use your intuition, which should work fine in 90% of cases.

Functions and operators are executed from high to low priority:

- High: power, square, cube, factorial, implied multiplication, functions
- Medium: multiplication and division
- Low: addition and subtraction, unary plus and minus

An operand with operators on both sides will associate with the higher priority operator of the two. If both priorities are the same, then the left operator (low and medium priority) or the right one (high priority) is applied first.

This order, following the customary "blackboard" notation, can be arbitrarily changed with parentheses. Of course, you may always do it when in doubt.

See also:

[Upper and lower case](#)

[Implied multiplication](#)

[Syntax restrictions](#)

[Examples](#)

Upper and lower case

Expressions do not have to be entered in a case-sensitive manner. If a lower case letter does not make sense in a given context, it will be raised to upper case (becoming a variable).

This is quite convenient, as we usually prefer to type just "2axsin3c" than "2 AX sin 3C".

Note, however, that "asin2x" will be understood as "asin 2X" and not "A sin 2X", as asin is recognized as a function name.

In case of doubt you may always add a separating space or use an upper case letter to have it understood as a variable.

Implied multiplication

The multiplication operator, \times , between two tokens (operands or symbols) can be skipped if

- the first one is a variable, a constant, a postfix operator or a closing parenthesis (except when used for function arguments), AND
- the second one is a variable, a symbolic constant, a function, or an opening parenthesis.

The difference between explicit and implied multiplication is that the latter has higher priority.

Examples:

Legal	Equivalent to	Illegal
$3AXZ$	$3 \times A \times X \times Z$	$AX3Z$
$2\pi C$	$2 \times \pi \times C$	$2 \ 3.14C$
$9 \sin X$	$9 \times \sin(X)$	$\sin(Y) \sin X$
$\ln X \exp Y$	$\ln(X \times \exp(Y))$	$\ln(X) \exp Y$
$\sin 2AB$	$\sin(2 \times A \times B)$	$\sin(2A) B$
$\sin X(A+B)$	$\sin(X \times (A+B))$	$\sin(A+B) X$
πZ^2	$\pi \times (Z^2)$	$Z^2 3.14$
$(A+B)(C+1)$	$(A+B) \times (C+1)$	$\ln(A+B)(C+1)$

The traditional notation is not quite consistent: "exp 3 ln X" is usually interpreted by a human reader as "exp(3×ln(X))" while "2 sin X cos X" as "2×sin(X)×cos(X)". Midget always chooses the first way, interpreting the latter example as "2×sin(X×cos(X))".

See also:

[Syntax restrictions](#)

Syntax restrictions

The "blackboard" notation is not always clear and consistent. Therefore, to avoid ambiguous expressions, we have to introduce a number of (hopefully, reasonable) restrictions.

- High-priority operators: 2 , 3 , $!$, $^{\wedge}$, implied multiplication, cannot follow a parenthesis enclosing a function argument (or arguments)

For example, the sequence " $\log(x)^2$ " is illegal; it should be changed into " $(\log x)^2$ " or " $\log(x^2)$ ", depending on the intended meaning. On the other hand, " $\log x^2$ " is OK and means " $\log(x^2)$ ".

- Sequences like " \sin^2x ", often used to denote " $(\sin x)^2$ ", are not recognized; the latter, explicit form has to be used.

See also:

[Implied multiplication](#)

[Examples](#)

Examples

$2*\sin 2x$	means	$2*\sin(2*x)$
$2 \sin 2*x$		$2*\sin(2) *x$
a^b^{xy}		$A^(B^(X*Y))$
$a+b*xy^{3.1}$		$A+(B*(X*(Y^{3.1})))$
$a \sin 2x$		$A*\sin(2*X)$
$\text{asin}2 x$		$\text{asin}(2*X)$
$\ln x^2$		$\ln(X^2)$

Variables

Variables are upper case letter symbols used to store numerical values.

- Midget has six "regular" variables, denoted as A, B, C, X, Y and Z (the registered version adds four more: D, P, Q and R).
- There is also the Temporary Variable, t (lower case!); it always denotes the result of the last computation (shown in the Result Display).
- The two symbolic constants, ¶ and é, behave in expressions exactly like variables, but they cannot be used to store calculation results. (To enter them, use Ctrl-p and Ctrl-e, respectively).

To store an expression value in a variable, type in the expression and press the "=" key; then select the variable from a pop-up menu. (Remember: a single numerical constant, like "1.25" is also an expression, although a very simple one!).

To use a variable in an expression, type it in, like in "1.25X-Z". The variable has to contain a previously stored value, otherwise an error will be reported.

To look up all variable values, press the "V" button.

Angular mode

The current angular mode is shown as [Rad] or [Deg] in a button to the right of the Result Display. Clicking on that button will change the mode.

Depending on this mode, Midget assumes that arguments to trigonometric functions are expressed in radians or degrees. It will also return results of inverse trigonometric functions in the same units.

Changing the angular mode to degrees can often lead to some confusion. I would recommend staying in the radian mode and using the angular constants explicitly. Then the calculation results can be displayed in the desired angular representation as needed.

See also:

[Angular constants](#)

Result format

Clicking on the Format button to the left of the Result Display offers (from a pop-up menu) a selection of various display formats.

- Default -- a reasonable compromise, the choice between floating and fixed point is made based on number readability. The values are shown with the full accuracy of 16 significant digits, and the trailing insignificant zeros are omitted.
- Fixed Point -- a fixed-point representation, with a selected number of digits after the decimal point (if the selected number would result in more than 16 significant digits, it will be appropriately reduced).
- Exponential -- a floating-point display, with a selected number of significant digits (up to 16) and at least two digits in the decimal exponent.

Any of these three settings becomes "permanent" -- it will also hold for all future calculation, until revoked. The other settings reformat just the current result.

- Fraction -- the value is shown as a whole part and a fraction, e.g., 3.125 becomes "3 1/8". The largest denominator is 1000 and it will be used if the value cannot be exactly represented with a smaller one.
- Deg-Min-Sec -- the value is shown in degrees, minutes and seconds of angle. This is affected by the current angular mode setting, Rad or Deg, as the result is assumed to be an angle, originally expressed in radians or degrees, respectively.
- Feet & Inches -- the result is assumed to be in feet and it will be shown as feet, inches and fraction (with the max. denominator of 64). For example, 1.24 will become "1 2 7/8" (one foot, two inches and 7/8 of an inch).
- Lbs & Ozs -- the result is treated as pounds and shown as pounds, ounces and fraction.
- Hrs-Min-Sec -- the result is treated as hours and shown as hours, minutes and seconds.

Obviously, depending on the displayed value and the format used, the display may be rounded. This never affects the actual result value, stored internally with 19 significant digits of accuracy.

Accuracy

The internal accuracy of individual arithmetic operations performed by Midget is 19 good decimal digits or better. These errors will, however, accumulate in computations involving multiple operations, so that 18 good digits is a safer assumption.

To be on the safe side, Midget displays only sixteen significant digits of results. In a great majority of cases all these digits will be valid with a reasonably large safety margin.

(Keep in mind, however, that for any finite floating-point accuracy, and for any calculator, one can easily invent an example of a computation where the relative result error will be 100% or more -- i.e., all decimal digits wrong.)

See also:

[Problems with the Win95 System Agent](#)

[The Pentium bug](#)

Problems with the Win95 System Agent

The System Agent, included with in the Microsoft's Win95 Plus, affects the accuracy of some floating-point calculations. Microsoft promises to fix the problem in one of the upcoming Windows 95 Service Packs.

In addition to reducing Midget's precision from 19 to about 16 digits, this may also affect reformatting the result as a fraction with use of the "Form" button.

My recommendation is to turn the System Agent off when using Midget (or any other math program). If the System Agent is present, you will see its icon in the tray at the right end of the launchpad.

To check for yourself, type in (notice twenty 3's after the decimal point):

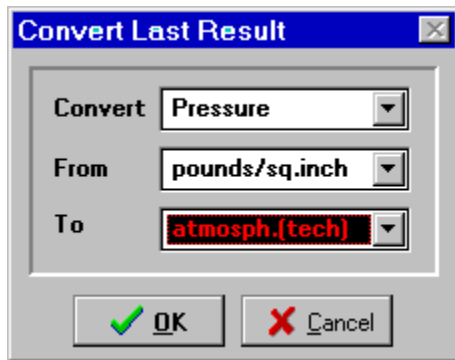
`1/3-.33333333333333333333`

and hit Enter (Midget will reformat the expression, truncating the decimal fraction from 20 to 18 digits). The displayed result should be near to 2E-20, but with the System Agent affecting the accuracy, it will be close to 2E-17..

The Pentium bug

The infamous bug in some of the early Pentium processors affects the accuracy of some division operations. Midget detects a faulty processor and uses additional code to work around the problem. (Really, the work-around has been built into the Delphi compiler by Borland.)

Unit conversion



The value shown in the Result Display can be converted from one kind of measurement units into another. This will affect not only the display, but also the actual value of the variable to which it refers.

Clicking on the [Convert] button will bring up the Unit Conversion Panel. It has three combo boxes with drop-down lists of items to choose from.

[Click on objects in the picture above to get more information on their purpose.](#)

Only the units relevant to the selected type will be available at any given moment; i.e. you will not be able to convert mass from feet to mph. (Hopefully, you will find this limitation not too painful...)

The actual conversion will take place when we exit the panel by clicking on [OK]. The variable storing the result will be affected by the conversion.

Note: although the temperature is not, strictly speaking, a physical quantity, temperature conversion has been also included.

See also:

[Result format](#)

The pop-up menu

This menu pops up when you click on the "?" button on the right. It groups some Midget settings and the less frequently accessed operations. Here are the choices:

- Auto scroll -- with this option checked, Midget will, after successfully completed operation, move to the next expression line, clearing it in the process.

(You can come back to the computed expression by scrolling back with the up-arrow key. The erased expression can be restored by hitting the Escape key before you started typing anything into the cleared line)

- Hints -- enables or disables the yellow fly-by hints. They can be useful in the beginning, but soon become annoying.
- Sound -- enables or disables the sounds (useful during corporate meetings or in the classroom)
- Help -- calls this help
- About Midget... -- displays the version and registration info
- Clear all -- clears all expressions and variables
- Exit -- exits the program.

Registered copies of Midget remember the last settings when the program is run again.

Enhanced features

Registered copies of Midget provide some enhanced features:

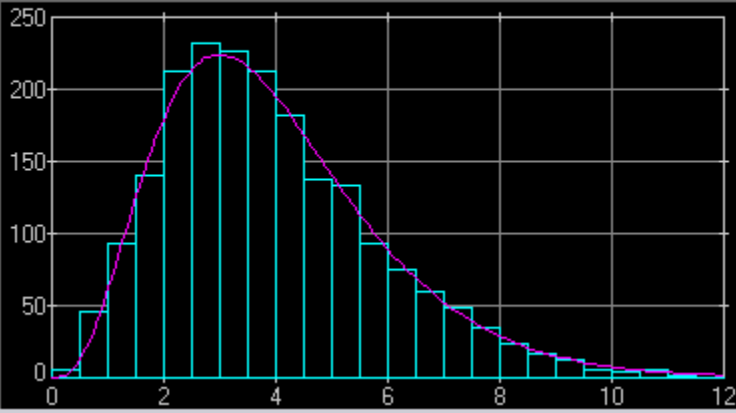
- The maximum expression length is 120 (not 44) characters
- Up to 24 (not 6) expressions are available at the same time
- Ten (not six) variables to store calculation results
- The program remembers last screen position and settings

Most importantly, you are paying me for the time I have spent to develop a product you use and enjoy (if you don't, just delete it from your system and forget this conversation). Thank you.

See also:

[How to register](#)

$.5 \cdot nn \cdot \text{fgam}(t, 4)$



= t ← Clr Deg 4

Vars Browser Param

```
nn: 2000.00000000  
mx: 3.92730550413  
sx: 1.930773694680  
vx: 3.727887060067  
x1: 0.274307444673  
x2: 16.18730564968
```

Add Hide Undo

Keyboard

Midget is run mostly from the keyboard (to provide buttons for all operators, functions, variables and digits would eat up lots of screen space). Here are the basics:

- The Delete, Backspace, left and right arrows, Home and End keys retain their customary Windows text-editing functionality.
- The up- and down-arrows are used to move between stored expressions.
- Escape (or Alt-Delete) clears the Display line with the current expression. If the line is already clear, Escape copies into it the last cleared expression.
- F1 calls this help
- F10 brings the pop-up menu
- F12 copies the displayed result into the Windows clipboard. To copy it into a document, switch to this document and press Shift-Ins
- The Control key is used, in keystroke combinations with other keys, to enter some special symbols onto the Expression Line.

There are also some less frequently accessed keystrokes:

- Alt-Insert copies the buffer into the current line. If the line is not empty, its old contents will be stored in the buffer (so that, effectively, a swap takes place).
- The Delete and Insert keys, when used together with Shift or Control, apply the standard Windows clip, copy and paste functions to the displayed expression. (Please refer to your Windows manual if in doubt.)

Unit conversion

Use this button to convert the result between various measurement units (e.g., from feet to centimeters).

The Expression Line

Here the expression you are typing in is being shown.
Hit the Enter key to compute the expression value.

Result format

Click here to change the format used to show the result.
You can change the number of digits, show the value as degrees, minutes and seconds, or as a fraction etc.

The Function Button

Pressing this button brings up a menu listing all available functions -- just as a reminder. Selecting from the menu may be also used instead of typing the function name.

Line number

Midget remembers up to six expressions (24 in the registered version).

The number of the currently shown expression is displayed here.

Clicking on this field will jump to the expression number one.

To move between expressions, use the up- and down-arrow keys.

The Menu Button

Clicking on this button will bring up the Midget pop-up menu with program options and some less frequently accessed operations.

Angular mode

This button changes the angular mode between Radians and Degrees. Depending on this mode Midget treats arguments to trigonometric functions as expressed in radians or degrees, and uses these units to return results of inverse trigonometric (arc) functions.

Result Display

Here the result of the last computation is shown. The result can be reformatted, or converted to different measurement units with use of the buttons on the left.

Click on this button to close the dialog box without doing anything.

Select the type of units, i.e. the physical quantity to convert (area, mass, etc.).

Select the units to convert from.

Click on this button to perform the conversion.

Select the units to convert to.

abs

abs X returns the absolute value of X

acos

acos X computes the arc cosine of X.

acot

acot X computes the arc cotangent of X.

Andrzej

Pronounced: "Un-jay". One of the most misspelled names east of Mississippi.

ang

ang(X,Y) computes the polar angle of the point (X,Y), equal to $\text{atan}(Y/X)$, and located in the proper quadrant

asin

asin X computes the arc sine of X.

atan

atan X computes the arc tangent of X.

Basic four operations

Note that "+" and "-" can be used as unary or binary operators, i.e. as in "-2*sin X" or as in "sin X - 4".

cos

cos X computes the cosine of X.

cot

cot X computes the cotangent of X.

cu

cu X computes the cube (third power) of X, same as the cube operator.

cur

cur X computes the cubic root of X.

exp

exp X computes the exponent of X, i.e. e^X .

Factorial

Defined for integer arguments as $X! = 1*2*...*X$.

For non-integer arguments it will be computed via the gamma function. (If you do not know what the gamma function is, just disregard this remark.)

fra

fra X returns the fractional part of X; for example, "fra 3.2" will return 0.2, and "fra(-3.2) will return -0.2

ft

ft(X,Y) converts X feet and Y inches to feet

hrs

hrs(X,Y,Z) converts X hours, Y minutes and Z seconds to hours

int

int X returns the integer part of X (rounded towards 0)

Kalkulator

About the name: this is an old Polish word; it can be translated into American English as "The Mother of All Calculators, and there ain't no better one, period".

Keystroke combinations

Ctrl-d enters the degree symbol, $^{\circ}$

Ctrl-p enters pi, π

Ctrl-e enters the Euler's e, shown as e

Ctrl-2 enters the square, 2

Ctrl-3 enters the cube, 3

lbs

lbs(X,Y) converts X pounds and Y ounces to pounds

ln

ln X computes the natural (base e) logarithm of X.

log

log X computes the decimal (common, base 10) logarithm of X.

max

max(X,Y) returns the greater of X and Y

MIDGET.KEY

This file should be put in the directory where all other Midget files are. It will activate all enhanced features of the program. It will be also recognized by all future versions of Midget (both 16- and 32-bit), so that you are getting free lifetime upgrades.

PLEASE REMEMBER TO KEEP A SAFE BACKUP OF THIS REGISTRATION FILE!!!!

min

`min(X,Y)` returns the lesser of X and Y

rad

rad(X,Y) computes the polar radius of the point (X,Y), equal to $\text{sqr}(X^2+Y^2)$

Raising to power

If the first operand of the power operator is negative, the second one has to have no fractional part.

For example, $(-3)^{1.5}$ will cause an error.

sgn

sgn X returns -1, 0 or 1 for negative, zero and positive X, respectively

sin

sin X computes the sine of X.

sq
sq X computes a square of X (same as the square operator)

sqr

sqr X computes a square root of X.

Square and cube

Use Ctrl-2 and Ctrl-3 keystroke combinations to enter these symbols.

tan

tan X computes the tangent of X.

ten

ten X computes ten to the power of X, i.e. 10^X .

Wrotniak

Ex-physicist, currently developing software for Air Traffic Control applications. Computer hobbyist, Chief Scientist and software engineer at Aerospace Engineering and Research in Maryland.

