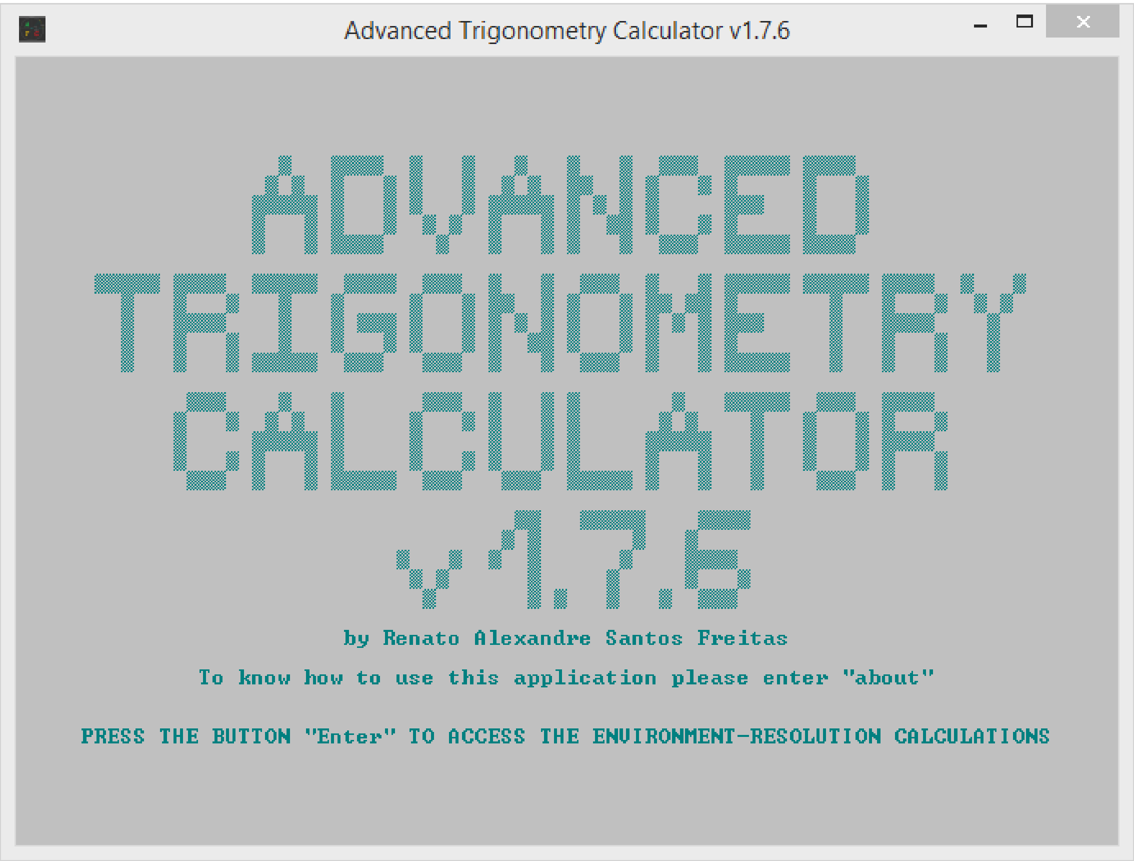
Advanced Trigonometry Calculator



```
ADVANCED
TRIGONOMETRY
CALCULATOR
v1.7.6

by Renato Alexandre Santos Freitas

To know how to use this application please enter "about"

PRESS THE BUTTON "Enter" TO ACCESS THE ENVIRONMENT-RESOLUTION CALCULATIONS
```

User Guide

All these application information contents were created and developed by Renato Alexandre dos Santos Freitas since March 19th, 2011 when this project was born.

Contents

Overview

Advanced Trigonometry Calculator is an open source calculator created in its entirety yet by Renato Alexandre dos Santos Freitas. In it you can find a calculator made primarily to solve trigonometric calculations but throughout its development acquired new functions beyond the trigonometric. Capabilities such as the introduction and response in four different numerical systems, the introduction and response in SI prefixes, the commands to force the trigonometric calculation mode, the creation and use of variables, the easy use of previous results, and the introduction of multiple expressions make this application stands out for its large capacities and its cost to the user.

Enabled functions

Trigonometry

cos(), acos(), sin(), asin(), tan(), atan(), sec(), asec(), cosec(), acosec(), cotan(), acotan()

As the application name says, it is an advanced calculator with focus in trigonometry, so it works correctly with all trigonometry calculations that you want to see solved.

Hyperbolic

cosh(), acosh(), sinh(), asinh(), tanh(), atanh(), sech(), asech(), cosech(), acosech(), cotanh(), acotanh()

As you may observe all trigonometry functions have a hyperbolic form and all these forms are also enabled in this application, working correctly.

Digital signal processing

sinc() (normalized sinc function)

Logarithm

log(), ln(), logb b()

You can use all logarithm bases that you want with the simple function “logb b()”. Just replace the space in "b b" by your base “2, 4, 8, etc.”

Arithmetic

rest, quotient, rtD D(), sqrt(), cbrt(), afact() and operators "+, -, *, /, ^, !"

You can use these functions to make advanced calculations. For rest and quotient functions you just need enter: dividend, function and divider, e.g. “100rest(3)=1” or “100quotient(3)=33”.

With “rtD D()” function you can use all root degrees that you want. Just replace the space in the function part "D D" by your degree “2, 3, 4, etc”. For the remaining functions: sqrt() is equivalent to rtD2D() and cbrt() is equivalent to rtD3D().

“afact()” is the inverse function of factorial operator “!”, e.g. “4!=24” so “afact(24)=4”.

The operators: addition, subtraction, multiplication, division, exponential, and factorial.

The negative character used is ‘_’, so to enter “-5” value, you need to enter “_5”.

Statistics

gerror(), gerrorinv(), gerrorc(), gerrorcinv(), qfunc(), qfuncinv()

"gerror()" is the error function also called "Gauss error function"; "qfunc()" is the Q-function. And in the functions above "inv" means inverse.

Commands

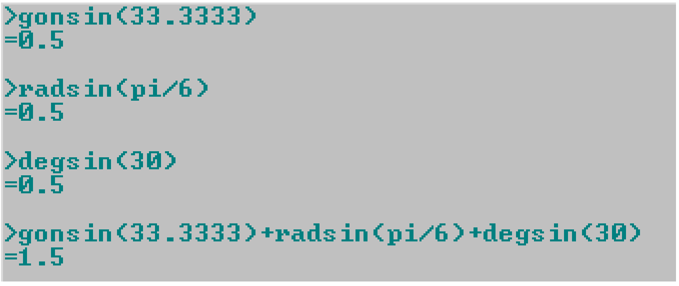
Commands	Action
clean	Let you clean the environment-resolution calculations window, if you enter many expressions with just one execution of this program, you will find it helpful.
ans	You can use this command in your entered expressions when you want do more calculations related with the last answer.
exit	Let you exit of application, closing the program.
about	Let you access the file “About execution of application.txt” that is inside the application. Avoiding you open this file. This file has information of all application features but less explained than in this user guide.
clean history	Let you clean the file “history.txt” with your entered expressions, respective answers and current time.
colors	Let you configure the text and background colors.
dimensions	Let you configure the dimensions, i.e. columns and lines numbers of environment-resolution calculations window.
window	Let you configure the position (X-axis, Y-axis), width, and height of application window.
mode	Let you choose what calculation mode you want to use in the trigonometric functions for the next expressions that you enter with trigonometric functions, by default is the degree mode until you configure another one.
see variables	Let you see the created variables with their values.
renamed variables	Let you see the created variables that need renamings.
eliminate variables	Let you eliminate all the created variables until this moment.
numerical systems	Let you enable or disable the functionality of show the answer for an expression entered, in the other three popular numerical systems, i.e.: binary, octal, and hexadecimal.
si prefixes	Let you enable or disable the functionality of show the answer for an expression entered, in the SI prefixes form, e.g. “1µ”.
actual time response	Let you enable or disable the functionality of show the current time after have been shown an answer for an expression entered.
time	Let you check the current time for a moment.
calendar	Let you check the current year calendar entering “calendar” or others if entered a year, e.g. “calendar(1991)”.
see results	Let you see the calculated results with their values.
eliminate results	Let you eliminate all the calculated results until this moment.
day of week	Let you check the day of week corresponding to a certain date, e.g. “dayofweek(d11m7y2014)” has as response “Friday”.
predefine txt	Let you predefine a “.txt” file for easily solve it later.
solve txt	Let you solve your predefined “.txt” file or others if you had created abbreviations for paths, e.g. “solvetxt(calculations)”
see abbreviations	Let you see the abbreviations created with their corresponding path.
eliminate abbreviations	Let you eliminate all abbreviations created until this moment.

These commands are the form of interact with this application, becoming it a trigonometry command prompt based application.

Useful features

Calculations mode

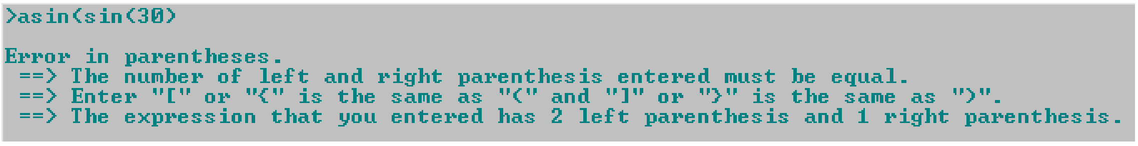
With the terms rad, deg, and gon you can force a trigonometry function to be calculated in radians, degrees, and gradians, respectively. So, although you have configured the calculations mode for trigonometry functions, you can use all of them in the same expression that you enter and get the correct answer. So, an example:



```
>gonsin(33.3333)
=0.5

>radsin(pi/6)
=0.5

>degsin(30)
=0.5

>gonsin(33.3333)+radsin(pi/6)+degsin(30)
=1.5
```

Parentheses

When entering complex expressions, you can use parentheses, by using these characters "(, [, {,),], }". So, an example of error message due to a bad use of parentheses:

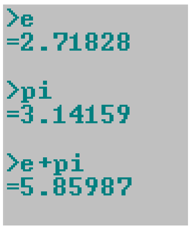


```
>asin(sin(30)

Error in parentheses.
 ==> The number of left and right parenthesis entered must be equal.
 ==> Enter "[" or "{" is the same as "(" and "]" or "}" is the same as ")".
 ==> The expression that you entered has 2 left parenthesis and 1 right parenthesis.
```

To help you, this application informs in this error message the number of right and left parentheses entered, in this case two left parentheses and one right parentheses.

Constants

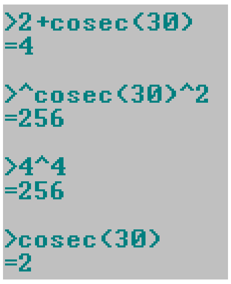
You may already know π and e constants but enter their values is boring, so you can enter these values by enter "e" and "pi". So, an example:



```
>e
=2.71828

>pi
=3.14159

>e+pi
=5.85987
```

Arithmetic operations

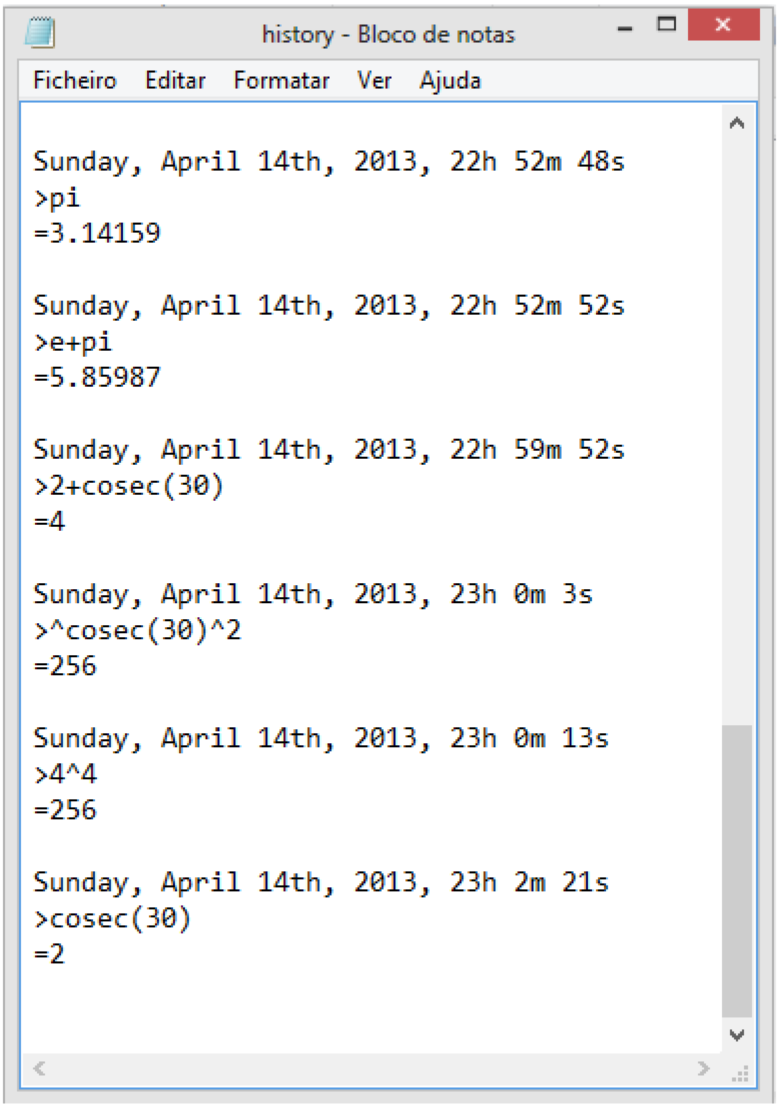
You can easily perform arithmetic operations with a previous and a current expression that you are entering using the characters “+, -, *, /, ^”. So, an example using degree mode:



```
>2+cosec(30)
=4

>^cosec(30)^2
=256

>4^4
=256

>cosec(30)
=2
```

History

To help you, this application creates a text file with the expressions that you entered, their answers and current time. Its name is "history.txt" you can find it inside the folder that contains the application executable in use. So, an example:

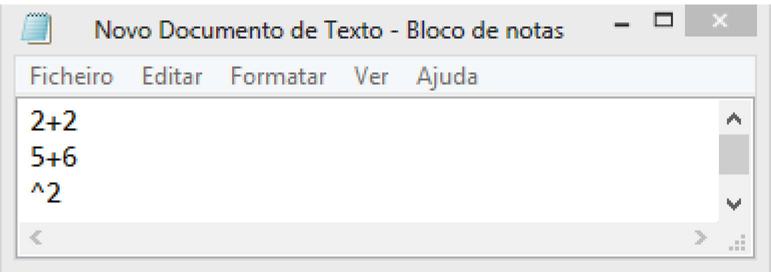


```
Sunday, April 14th, 2013, 22h 52m 48s
>pi
=3.14159

Sunday, April 14th, 2013, 22h 52m 52s
>e+pi
=5.85987

Sunday, April 14th, 2013, 22h 59m 52s
>2+cosec(30)
=4

Sunday, April 14th, 2013, 23h 0m 3s
>^cosec(30)^2
=256

Sunday, April 14th, 2013, 23h 0m 13s
>4^4
=256

Sunday, April 14th, 2013, 23h 2m 21s
>cosec(30)
=2
```

Processing of text files (.txt)

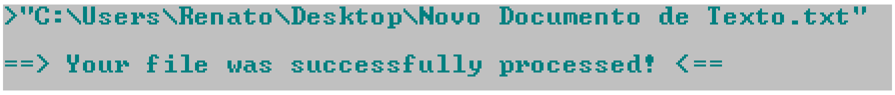
If you create a ".txt" file with an expression per line, saving the file. And dragging the icon of the file to the environment-resolution calculations and pressing "Enter" button the application reads the file expressions and gives to them an answer. Creating a new file with the expressions and their answers. This file has the same name of the original but terminating in "_answers". So, examples:

Original file:



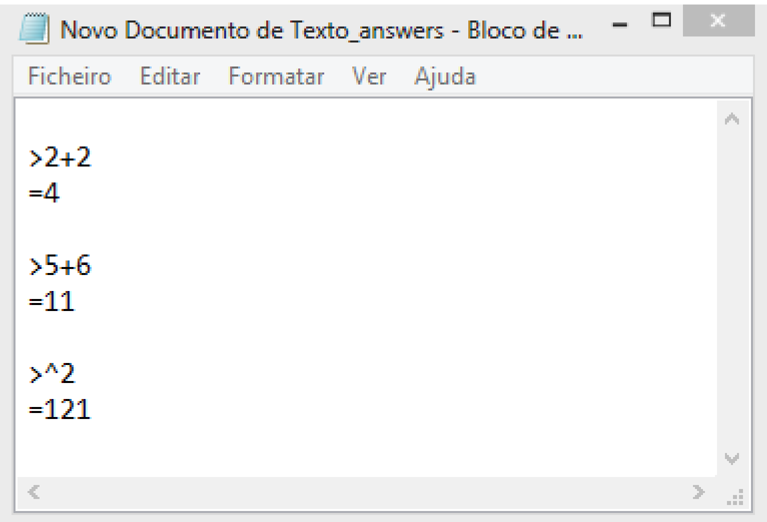
```
2+2
5+6
^2
```

Interaction with application:



```
>"C:\Users\Renato\Desktop\Novo Documento de Texto.txt"
==> Your file was successfully processed! <==
```

New file with answers:



```

>2+2
=4

>5+6
=11

>^2
=121
```

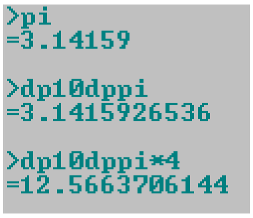
You can also predefine your ".txt" file and solve it easily, using "predefinetxt" and "solvetxt". Check commands page for details.

User configurations

This application lets you configure the dimensions, the colors, the window, and the mode. If you want reset these configurations, you just need close the application, go to the folder that contains the executable file in use of the application and eliminate all the ".txt" files less "About execution of application.txt" and "License.txt".

Number of decimal places

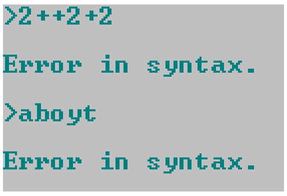
You can choose how many decimal places you want that exist when the answer is displayed. You just need enter "dpnumberdp" and the expression that you want calculated. So, an example:



```
>pi
=3.14159

>dp10dppi
=3.1415926536

>dp10dppi*4
=12.5663706144
```

Verification of entered data

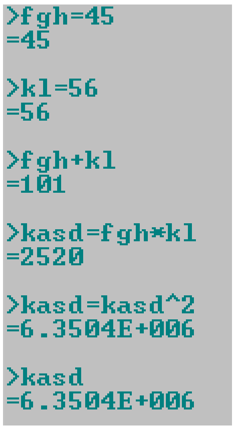
There is a high capacity of verification of entered data to detect entered errors, like commands badly entered, e.g. “aboyt” instead of “about”, double arithmetic operator entered, e.g. “2++2+2” instead of “2+2+2”, an arithmetic operator entered in the end of an expression e.g. “2sin(30)+”, and variables and functions names badly entered. It works like a high performance of data verification feature.



```
>2++2+2

Error in syntax.

>aboyt

Error in syntax.
```

Variables

You can create your variables and use them in the next entered expressions. So, an example:

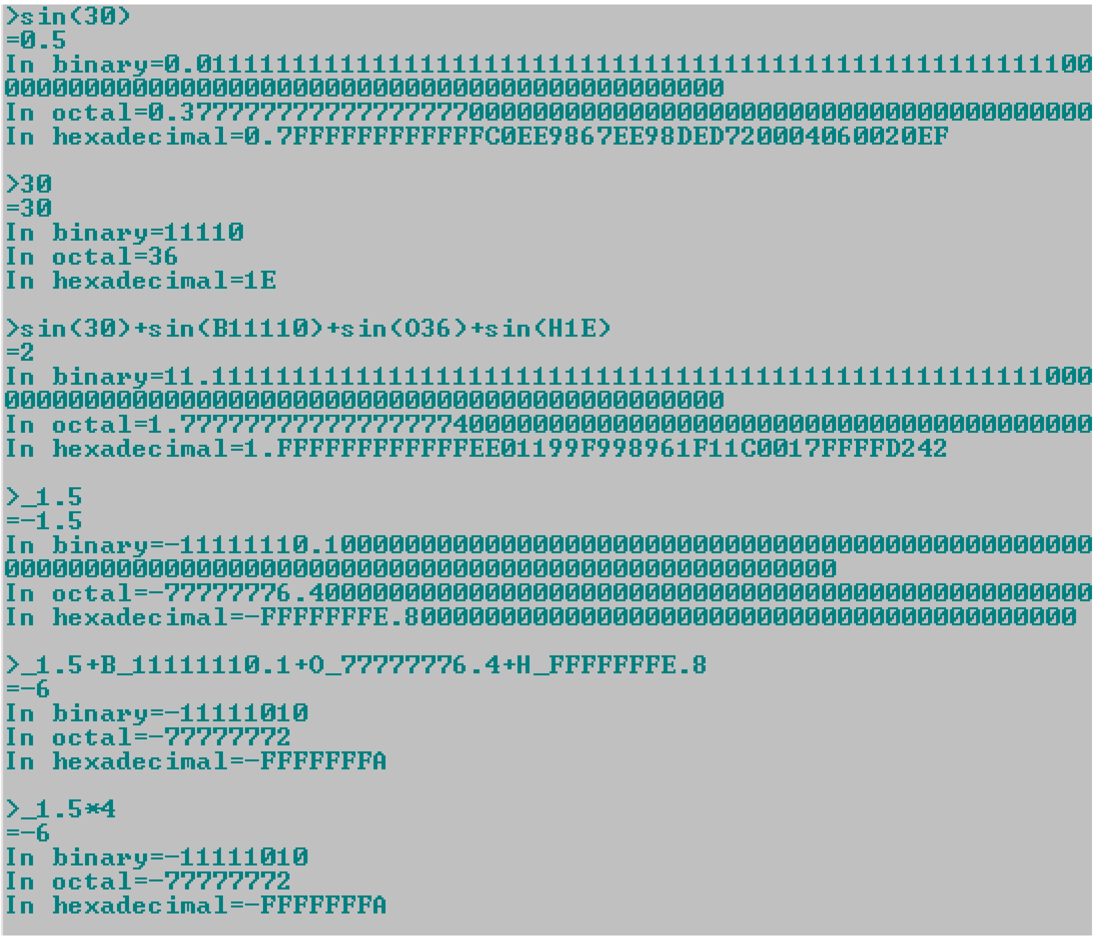


```
>fgh=45
=45

>kl=56
=56

>fgh+kl
=101

>kasd=fgh*kl
=2520

>kasd=kasd^2
=6.3504E+006

>kasd
=6.3504E+006
```

As you may notice the creation and use of variables is a cool feature that allows an easier use of results of expressions already solved. And so, you have the possibility of create expressions easier than with just numeric digits and functions.

Numerical Systems

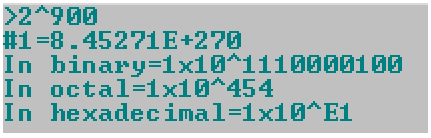
You can enter your expression values in four different numerical systems. So, an example:



```
>sin(30)
=0.5
In binary=0.01111111111111111111111111111111111111111111111111111100
000000000000000000000000000000000000000000000
In octal=0.377777777777777777000000000000000000000000000000000000000
In hexadecimal=0.7FFFFFFFFFFFFC0EE9867EE98DED720004060020EF

>30
=30
In binary=11110
In octal=36
In hexadecimal=1E

>sin(30)+sin(B11110)+sin(O36)+sin(H1E)
=2
In binary=11.1111111111111111111111111111111111111111111111111111000
000000000000000000000000000000000000000000000
In octal=1.77777777777777777400000000000000000000000000000000000000
In hexadecimal=1.FFFFFFFFFFFEE01199F998961F11C0017FFFFD242

>_1.5
=-1.5
In binary=-11111110.100000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
In octal=-77777776.4000000000000000000000000000000000000000000000000
In hexadecimal=-FFFFFFFE.800000000000000000000000000000000000000000

>_1.5+B_11111110.1+O_77777776.4+H_FFFFFFFE.8
=-6
In binary=-11111010
In octal=-77777772
In hexadecimal=-FFFFFFFA

>_1.5*4
=-6
In binary=-11111010
In octal=-77777772
In hexadecimal=-FFFFFFFA
```

B - Binary, O - Octal, H - Hexadecimal

You can get your numerical systems answer in scientific notation. So, an example:



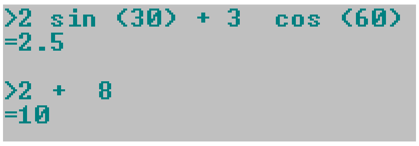
```
>2^900
#1=8.45271E+270
In binary=1x10^1110000100
In octal=1x10^454
In hexadecimal=1x10^E1
```

Reuse of entered expressions

You can use the arrow keys "up" and "down" of keyboard to reuse expressions that you have already entered, you can edit them, and so, get the wanted answer in a rapid and simple form.

Use of the space key

Feel free to use the key of "space" of your keyboard. If you like to use some space when you are entering an expression, stay know that you can do it freely, just in the environment-resolution calculations. So, an example:

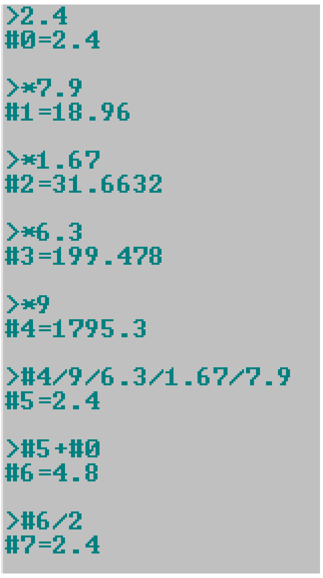


```
>2 sin (30) + 3  cos (60)
=2.5

>2 +  8
=10
```

As you notice the use of the space key don't affects the correct processing of the expressions.

Use of previous results

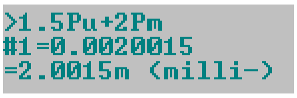
You can use previous results easily because they have a name. So, an example:



```
>2.4
#0=2.4

>*7.9
#1=18.96

>*1.67
#2=31.6632

>*6.3
#3=199.478

>*9
#4=1795.3

>#4/9/6.3/1.67/7.9
#5=2.4

>#5+#0
#6=4.8

>#6/2
#7=2.4
```

SI prefixes

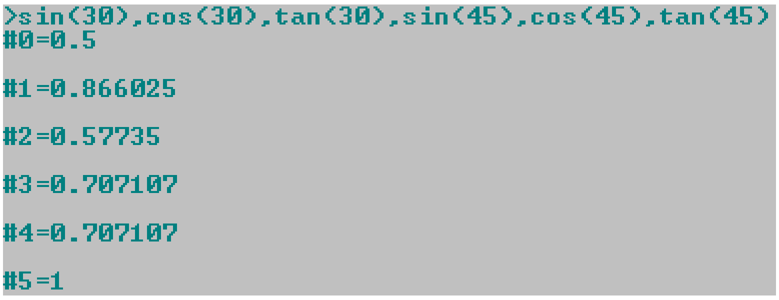
You can enter your values in a SI prefixes manner, to it be solved, you need enter a 'P' previously of prefix, because 'P' says to the application that you are entering a value in the SI prefixes manner. To enter the value "2p", you need to enter "2Pp". The prefixes replace the need of scientific notation.



```
>1.5Pu+2Pm
#1=0.0020015
=2.0015m (milli-)
```

Introduction of multiple expressions

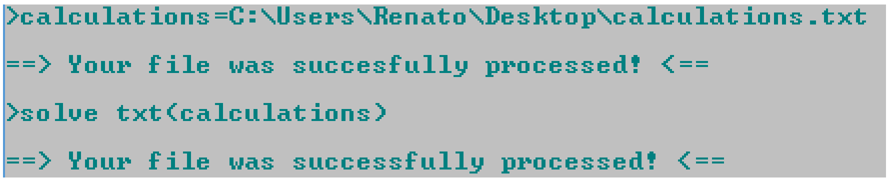
You can enter multiple expressions by pressing “Enter” button just a time. For it you need separate your expressions using "," (comma). Take an example:



```
>sin(30),cos(30),tan(30),sin(45),cos(45),tan(45)
#0=0.5

#1=0.866025

#2=0.57735

#3=0.707107

#4=0.707107

#5=1
```

Creating abbreviations for paths

You can make your work simpler if you create abbreviations for paths. Take an example:



```
>calculations=C:\Users\Renato\Desktop\calculations.txt

==> Your file was succesfully processed! <==

>solve txt(calculations)

==> Your file was successfully processed! <==
```

The first expression defines your abbreviation and processes the “.txt” file. The second expression processes the “.txt” file that abbreviation corresponds to.

Download

Setup Advanced Trigonometry Calculator.exe – installation file.

Advanced Trigonometry Calculator.zip – portable file.

Support the development of "Advanced Trigonometry Calculator" by donating if you find it useful!

To donate, click on the image below.

Give your feedback contacting the developer!

Via facebook application's page we can talk about it, give a "like" in the page. Click in the facebook logo image below.

Email: renato_freitas91@hotmail.com
Mobile phone: 00351967886392

Developer

Renato Alexandre dos Santos Freitas is Portuguese, he has a degree in Electrical and Telecommunication Engineering by Escola Superior de Tecnologia de Castelo Branco, part of Polytechnic Institute of Castelo Branco in Portugal.

Renato Freitas was born in July 6th, 1991 in Coimbra, Portugal and lived his growth, after his 3 years old, in Castelo Branco, Portugal. To you have an idea, he likes walking, cycling, football, swimming, programming, studying, make friends, socialize, watch documentaries of scientific and technological nature, attend lectures of scientific and technological nature, discover new things, and he searches every time for more knowledge.