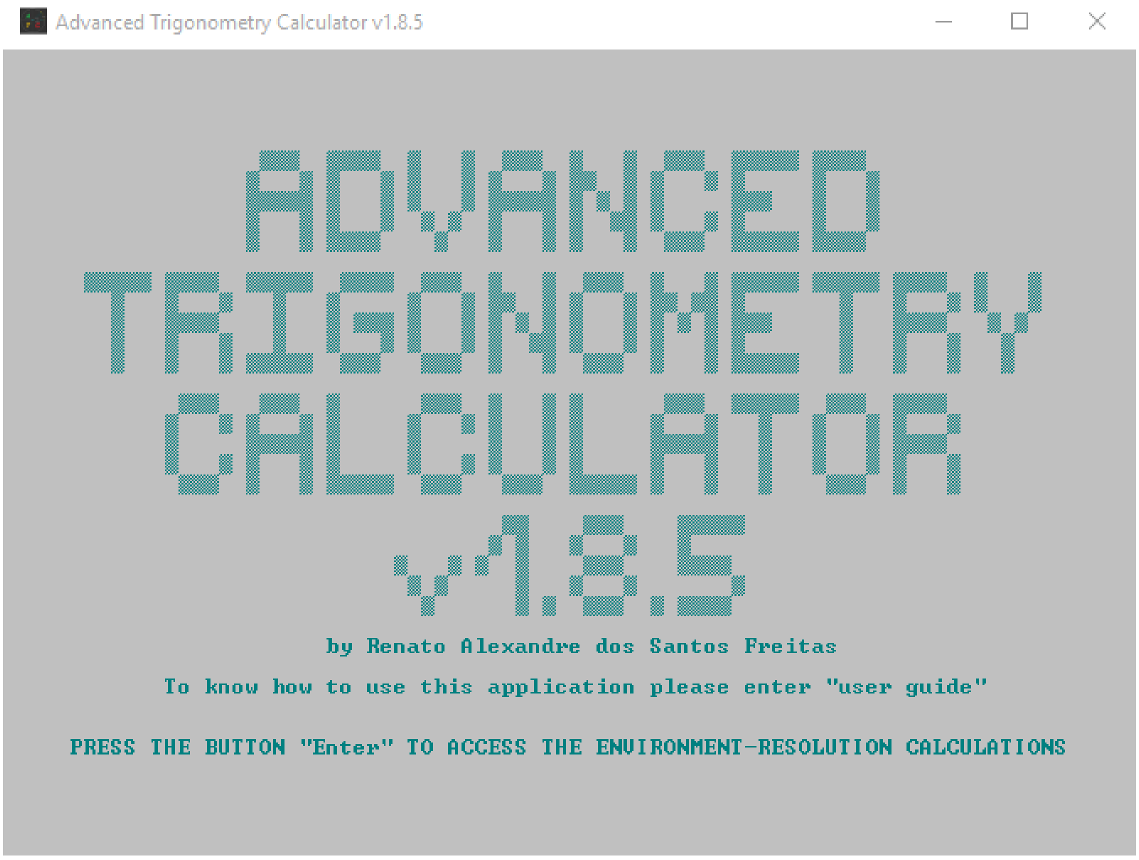
# Advanced Trigonometry Calculator



```
ADVANCED
TRIGONOMETRY
CALCULATOR
v1.8.5

by Renato Alexandre dos Santos Freitas

To know how to use this application please enter "user guide"

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 a rock-solid calculator, allowing you perform advanced complex math calculations.

Enter your complex math expression on its integrity and in the final press “Enter” button, after some instants the solution for your expression will be displayed.

Anyone can use this calculator since the syntax used is very similar with scientific handheld calculators, e.g. TI 84-Plus.

# Enabled functions

## Trigonometry

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

This calculator can perform trigonometric calculations with complex numbers as arguments.

## Hyperbolic

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

This calculator can perform hyperbolic calculations with complex numbers as arguments.

## Digital signal processing

sinc() (normalized sinc function)

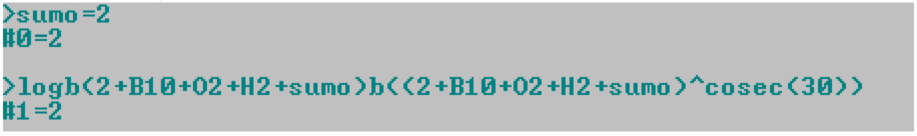
This function can perform calculations with complex numbers as argument.

## Logarithm

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

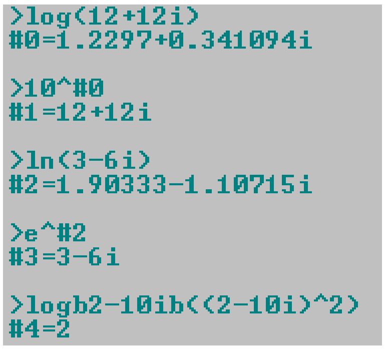
You can use all logarithm bases that you want with the simple function “logb b()”. Just replace the space on "b b" by your base “2, 4, 8, etc.” You can also use complex numbers as logarithm base.

You can put between “b b” simple arithmetic calculations but please don’t put functions. Example: “logb(2+B10+O2+H2+sumo)b((2+B10+O2+H2+sumo)^cosec(30))”



```
>sumo=2
#0=2

>logb(2+B10+O2+H2+sumo)b((2+B10+O2+H2+sumo)^cosec(30))
#1=2
```

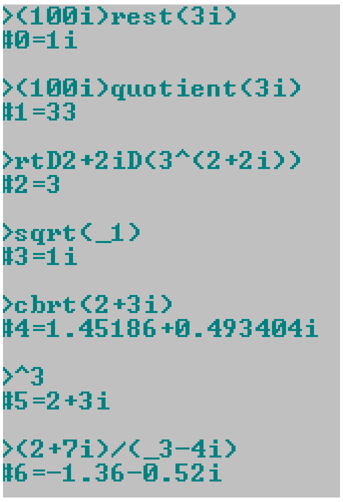
Example of complex numbers applying to logarithm functions:



```
>log(12+12i)
#0=1.2297+0.341094i

>10^#0
#1=12+12i

>ln(3-6i)
#2=1.90333-1.10715i

>e^#2
#3=3-6i

>logb2-10ib((2-10i)^2)
#4=2
```

## Arithmetic

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

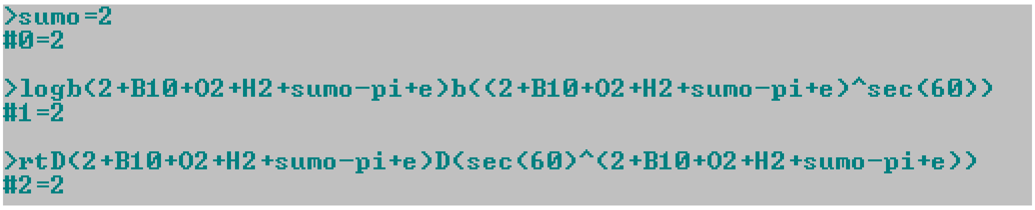


```
>(100i)rest(3i)
#0=1i

>(100i)quotient(3i)
#1=33

>rtD2+2iD(3^(2+2i))
#2=3

>sqrt(_1)
#3=1i

>cbrt(2+3i)
#4=1.45186+0.493404i

>^3
#5=2+3i

>(2+7i)/(_3-4i)
#6=-1.36-0.52i
```

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().

You can put between “D D” simple arithmetic calculations but please don’t put functions. Example: “rtD(2+B10+O2+H2+sumo-pi+e)D(sec(60)^(2+B10+O2+H2+sumo-pi+e))”



```
>sumo=2
#0=2

>logb(2+B10+O2+H2+sumo-pi+e)b((2+B10+O2+H2+sumo-pi+e)^sec(60))
#1=2

>rtD(2+B10+O2+H2+sumo-pi+e)D(sec(60)^(2+B10+O2+H2+sumo-pi+e))
#2=2
```

“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 on 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 have automatic renamings for correct processing. |
| 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. |

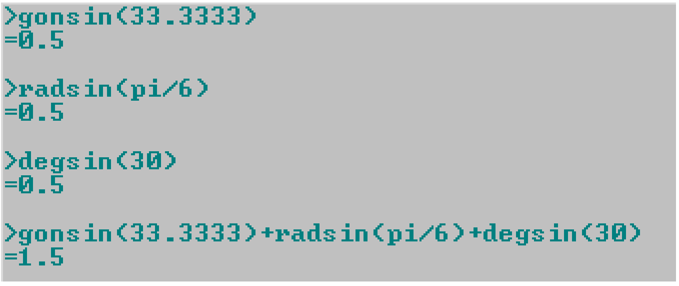
| | |
|---|---|
| solve equations system | Let you solve equations systems. Example of use: "solve equations system(2\4\9;5\6\12)" resulting on the solutions "x1=-0.75" and "x2=2.625". Check the section “Features”. |
| stopwatch | Let you  measure how much time you spend for anything. Entering “stopwatch(3)” let you mark 3 times. To mark times you just need press the button “Enter”. Check the section “Features”. |
| timer | Let you control the time you spend for anything. Entering “timer(0:5:0)” you will be notified when passed 5 minutes since the press of button “Enter”. Check the section “Features”. |
| clock | Let you use a clock. Entering "clock(0:5:0)" you will have a clock during 5 minutes. Check the section “Features”. |
| user guide | Let you open this user guide. |
| update | Let you download the latest version of the application. |
| update portable | Let you download the latest portable version of the application. |
| run atc | Let you use the application on multiple execution. When you enter this command you ask to run the application executable one more time, and you can keep doing it so on. |
| shutdown | Let you shutdown your PC. |
| restart | Let you restart your PC. |
| hibernate | Let you hibernate your PC. Administrator privileges are needed. |
| log off | Let you log off your PC. |
| sleep | Let you sleep your PC. Administrator privileges are needed. |
| lock | Let you lock your PC. |
| reset all | Let you delete all application ".txt" files, less the files of application info and application license. Application will be as it had finished installation or portable version had finished its unpacking. |
| reset settings | Let you delete only the files that were created due to configurations in the application made by the user. |
| restart atc | Let you close the application and then execute it (restart app). |
| history | Let you open history file that has your application use history. |
| atc folder | Let you open the application folder. |
| source code | Let you open the source code folder. |
| to solve | Let you open the folder with the txt files for txt detector feature. Check the section “Features”. |
| enable txt detector | Let you enable again the txt detector feature. Check the section “Features”. |

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

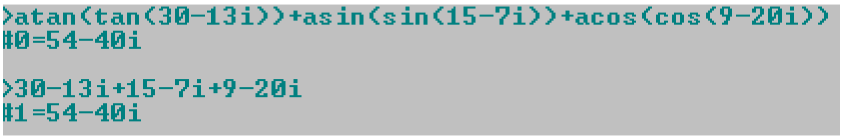
# Features

## 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 are entering and get the correct answer. So, an example:



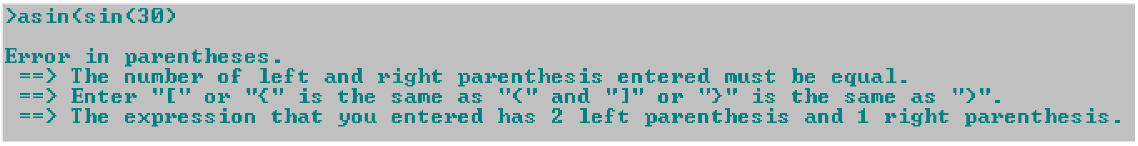
```
>gonsin(33.3333)
=0.5

>radsin(pi/6)
=0.5

>degsin(30)
=0.5

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



```
>atan(tan(30-13i))+asin(sin(15-7i))+acos(cos(9-20i))
#0=54-40i

>30-13i+15-7i+9-20i
#1=54-40i
```

## 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:

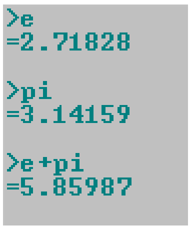


```
>asin(sin(30)

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.
```

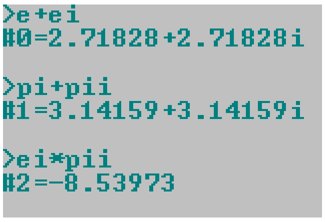
To help you, this application informs in this error message the number of right and left parentheses entered, in this case two left parentheses and one right parentheses.

## Constants

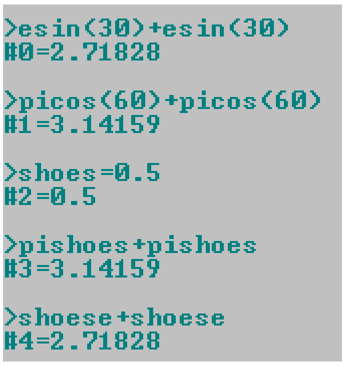
You may already know $\pi$ and $e$ constants but enter their values is boring, so you can enter these values by enter “e” and “pi”. So, two examples:



```
>e
=2.71828

>pi
=3.14159

>e+pi
=5.85987
```



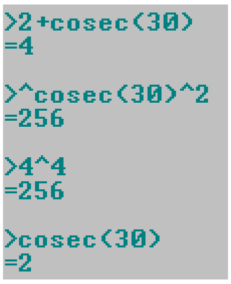
```
>e+ei
#0=2.71828+2.71828i

>pi+pii
#1=3.14159+3.14159i

>ei*pii
#2=-8.53973
```



```
>esin(30)+esin(30)
#0=2.71828

>picos(60)+picos(60)
#1=3.14159

>shoes=0.5
#2=0.5

>pishoes+pishoes
#3=3.14159

>shoese+shoese
#4=2.71828
```

## Arithmetic operations

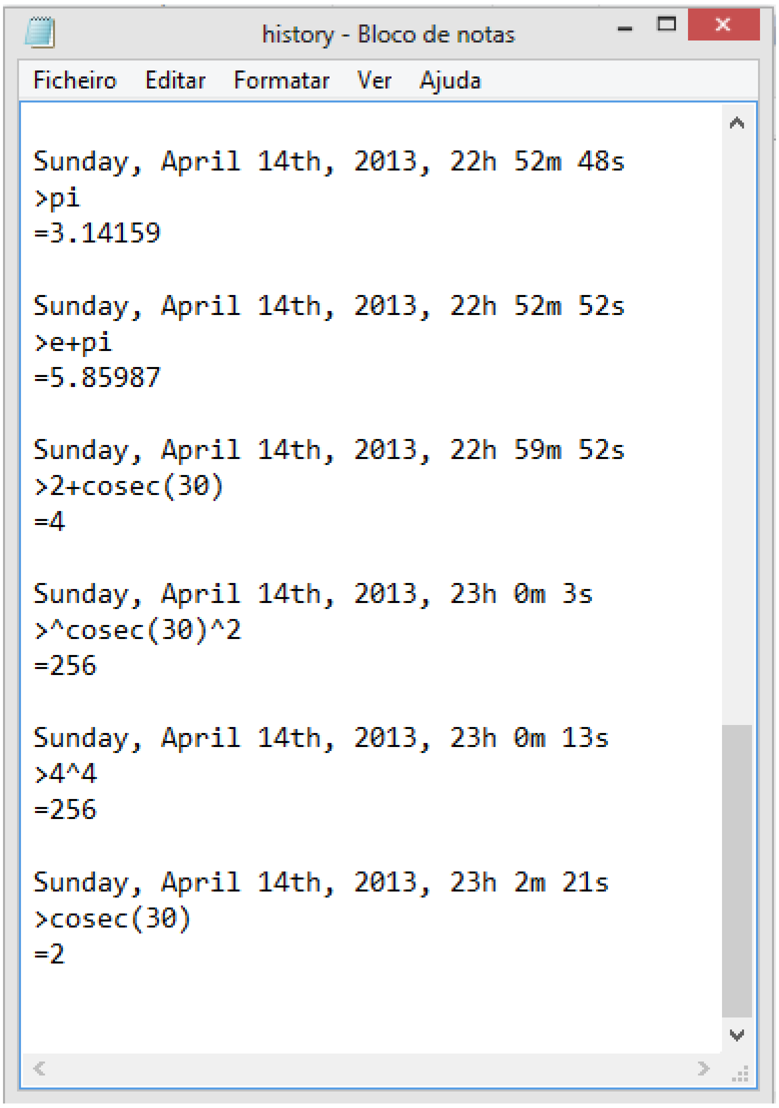
You can easily perform arithmetic operations with a previous and a current expression that you are entering, using the characters “+, -, *, /, ^”. So, an example using degree mode:



```
>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 have 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. You can also enter “history” and see the file automatically. So, an example:

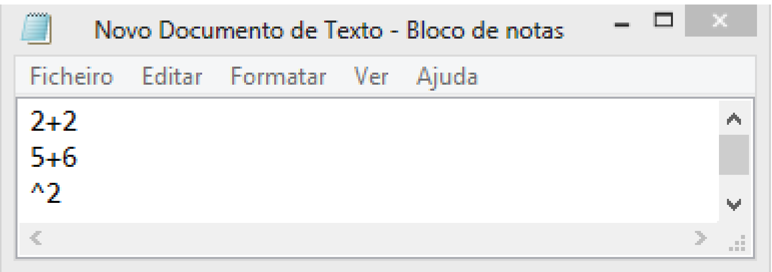


```
Sunday, April 14th, 2013, 22h 52m 48s
>pi
=3.14159

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)

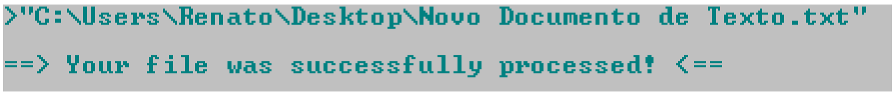
If you create a ".txt" file with expressions separated by ',' or by paragraphs (by pressing "Enter" button), saving the file. Dragging the icon of the file to the environment-resolution calculations and pressing "Enter" button, the application reads the file expressions and gives to them an answer. Creating a new file with the expressions and their answers. This file has the same name of the original but terminates on "_answers". So, examples:

Original file:


Novo Documento de Texto - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda

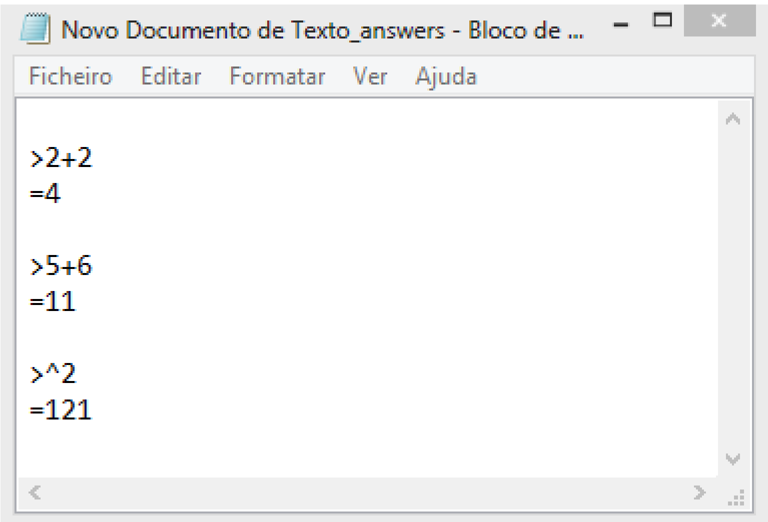
```
2+2
5+6
^2
```

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

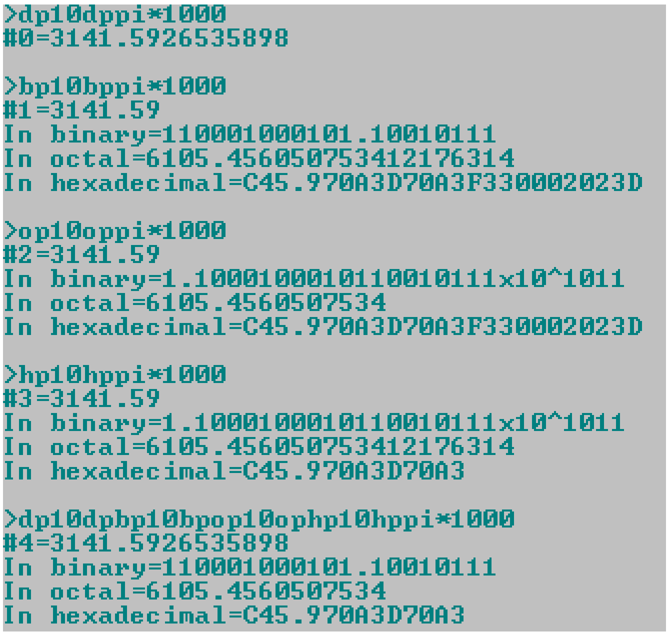
This application lets you configure the dimensions, the colors, the window, and the mode. If you want reset these configurations:

- Enter "reset all" to put app as it had finished its installation or it had finished its unpacking
- Enter "reset settings" to put the configurations by default.

For both commands you need to restart application. You can do it by enter "restart atc".

## Number of decimal, binary, octal and hexadecimal places

You can choose how many decimal, binary, octal and hexadecimals places you want that exist when the answer is displayed. For example, you just need enter "dpnumberdp" and the expression that you want be calculated to define the number of decimal places. So, an example:



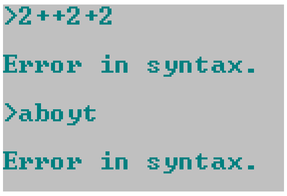
```
>dp10dppi*1000
#0=3141.5926535898

>bp10bppi*1000
#1=3141.59
In binary=110001000101.10010111
In octal=6105.456050753412176314
In hexadecimal=C45.970A3D70A3F330002023D

>op10oppi*1000
#2=3141.59
In binary=1.1000100010110010111x10^1011
In octal=6105.4560507534
In hexadecimal=C45.970A3D70A3F330002023D

>hp10hppi*1000
#3=3141.59
In binary=1.1000100010110010111x10^1011
In octal=6105.456050753412176314
In hexadecimal=C45.970A3D70A3

>dp10dpbp10bpop10ophp10hppi*1000
#4=3141.5926535898
In binary=110001000101.10010111
In octal=6105.4560507534
In hexadecimal=C45.970A3D70A3
```

dp → decimal places
bp → binary places
op → octal places
hp → hexadecimal places

Note that numerical systems response is used to give you the answer. You don't have to enable it in this case because the application detects that this feature is needed and use it automatically.

## Verification of entered data

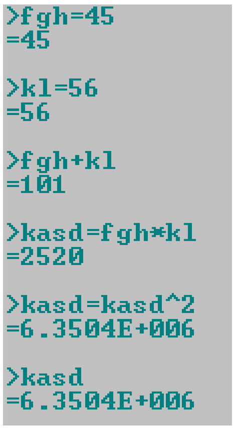
There is a high capacity of verification of entered data to detect entered errors, like commands badly entered, e.g. "aboyt" instead of "about", double arithmetic operator entered, e.g. "2++2+2" instead of "2+2+2", an arithmetic operator entered at the end of an expression e.g. "2sin(30)+", and variables 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 on the next entered expressions. So, an example:

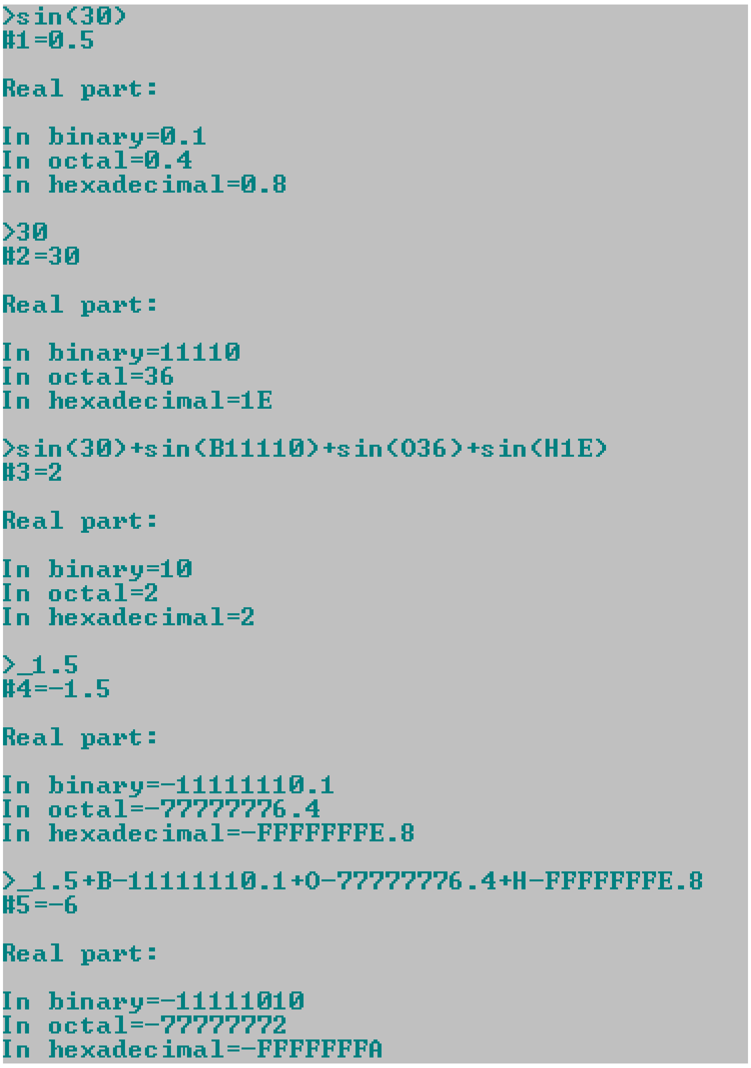


```
>fgh=45
=45

>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

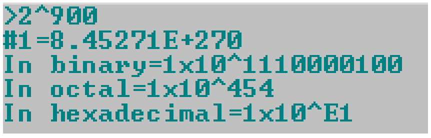
You can enter your expression values in four different numerical systems. So, an example:



```
>sin(30)
#1=0.5

Real part:

In binary=0.1
In octal=0.4
In hexadecimal=0.8

>30
#2=30

Real part:

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

>sin(30)+sin(B11110)+sin(O36)+sin(H1E)
#3=2

Real part:

In binary=10
In octal=2
In hexadecimal=2

>_1.5
#4=-1.5

Real part:

In binary=-11111110.1
In octal=-77777776.4
In hexadecimal=-FFFFFFFE.8

>_1.5+B-11111110.1+O-77777776.4+H-FFFFFFFE.8
#5=-6

Real part:

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:



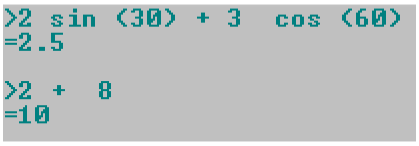
```
>2^900
#1=8.45271E+270
In binary=1x10^1110000100
In octal=1x10^454
In hexadecimal=1x10^E1
```

## 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 way.

## 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 on the environment-resolution calculations. So, an example:

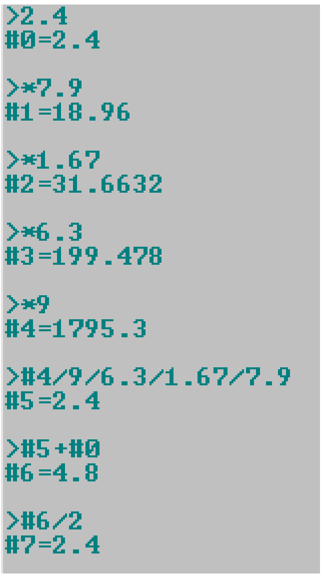


```
>2 sin (30) + 3  cos (60)
=2.5

>2 +  8
=10
```

As you notice the use of the space key doesn't affect the correct processing of the expressions.

## Use of previous results

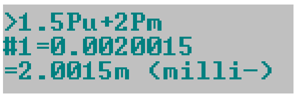
You can use previous results easily because they have a name. So, an example:



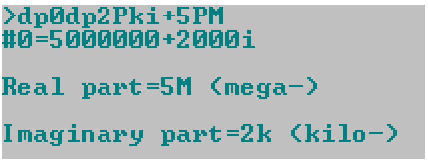
```
>2.4
#0=2.4

>*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, for it you need enter a 'P' previously of prefix, because 'P' says to the application that you are entering a value in a 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-)
```



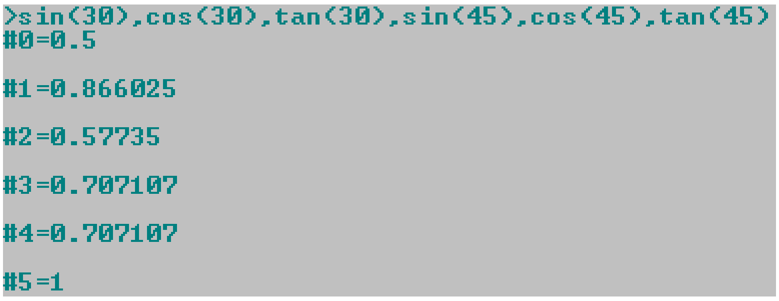
```
>dp0dp2Pki+5PM
#0=5000000+2000i

Real part=5M (mega-)

Imaginary part=2k (kilo-)
```

The table below has the equivalence between Prefix SI and its corresponding value.

| Way of putting Prefix SI | Equivalent value |
|---|---|
| PY | 1E+24 |
| PZ | 1E+21 |
| PE | 1E+18 |
| PP | 1E+15 |
| PT | 1E+12 |
| PG | 1E+9 |
| PM | 1E+6 |
| Pk | 1000 |
| Ph | 100 |
| Pda | 10 |
| Pd | 0.1 |
| Pc | 0.01 |
| Pm | 0.001 |
| Pu | 1E-6 |
| Pn | 1E-9 |
| Pp | 1E-12 |
| Pf | 1E-15 |
| Pa | 1E-18 |
| Pz | 1E-21 |
| Py | 1E-24 |

## Introduction of multiple expressions

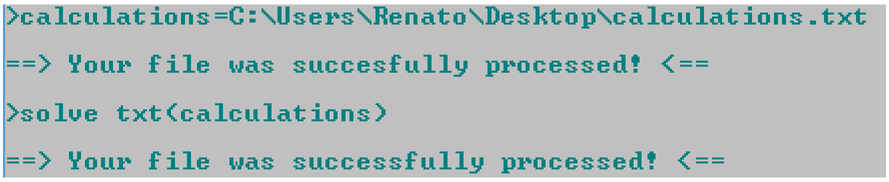
You can enter multiple expressions by pressing “Enter” button just a time. For it you need separate your expressions using "," (comma). Take an example:



```
>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 become 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! <==
```

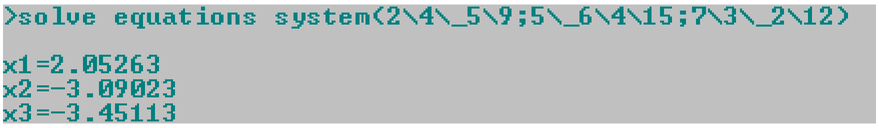
The first expression defines your abbreviation and processes the “.txt” file. The second expression processes the “.txt” file that abbreviation corresponds to.

## Solving equations systems

You can solve equations systems. Take an example:

$$\begin{cases} 2x + 4y - 5z = 9 \\ 5x - 6y + 4z = 15 \\ 7x + 3y - 2z = 12 \end{cases}$$

$$\begin{array}{cccc} 2 & 4 & -5 & 9 \\ 5 & -6 & 4 & 15 \\ 7 & 3 & -2 & 12 \end{array}$$

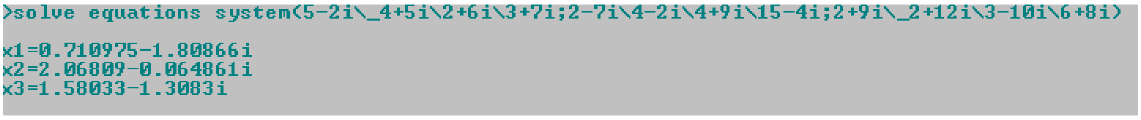


```
>solve equations system(2\4\_5\9;5\_6\4\15;7\3\_2\12)

x1=2.05263
x2=-3.09023
x3=-3.45113
```

As you can observe, '\' character separates the columns and ';' the rows.

$$\begin{cases} (5 - 2i)x + (-4 + 5i)y + (2 + 6i)z = 3 + 7i \\ (2 - 7i)x + (4 - 2i)y + (4 + 9i)z = 15 - 4i \\ (2 + 9i)x + (-2 + 12i)y + (3 - 10i)z = 6 + 8i \end{cases}$$

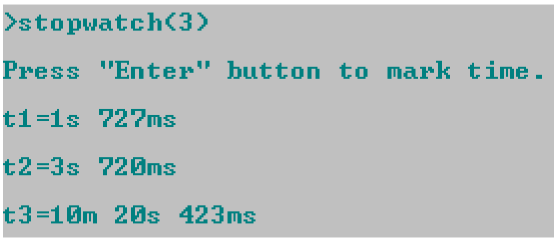
$$\begin{array}{cccc} 5 - 2i & -4 + 5i & 2 + 6i & 3 + 7i \\ 2 - 7i & 4 - 2i & 4 + 9i & 15 - 4i \\ 2 + 9i & -2 - 12i & 3 - 10i & 6 + 8i \end{array}$$



```
>solve equations system(5-2i\_4+5i\2+6i\3+7i;2-7i\4-2i\4+9i\15-4i;2+9i\_2+12i\3-10i\6+8i)

x1=0.710975-1.80866i
x2=2.06809-0.064861i
x3=1.58033-1.3083i
```

## Stopwatch

Try measure the time you spend doing something.

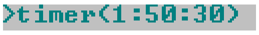


```
>stopwatch(3)
Press "Enter" button to mark time.
t1=1s 727ms
t2=3s 720ms
t3=10m 20s 423ms
```

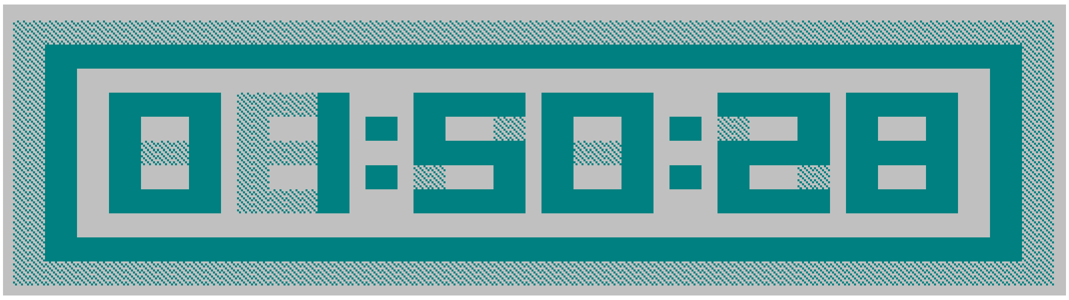
Your measure can go until days, e.g. "1d 5h 23m 34s 126ms".

## Timer

Manage your time.



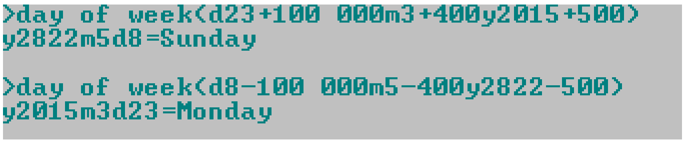
```
>timer(1:50:30)
```



```
01:50:28
```

In 1 hour, 50 minutes and 28 seconds, you would be notified that time has finished. The notification is a beep that is audible during 20 seconds.

## Date adjustment

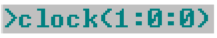


```
>day of week(d23+100 000m3+400y2015+500)
y2822m5d8=Sunday

>day of week(d8-100 000m5-400y2822-500)
y2015m3d23=Monday
```

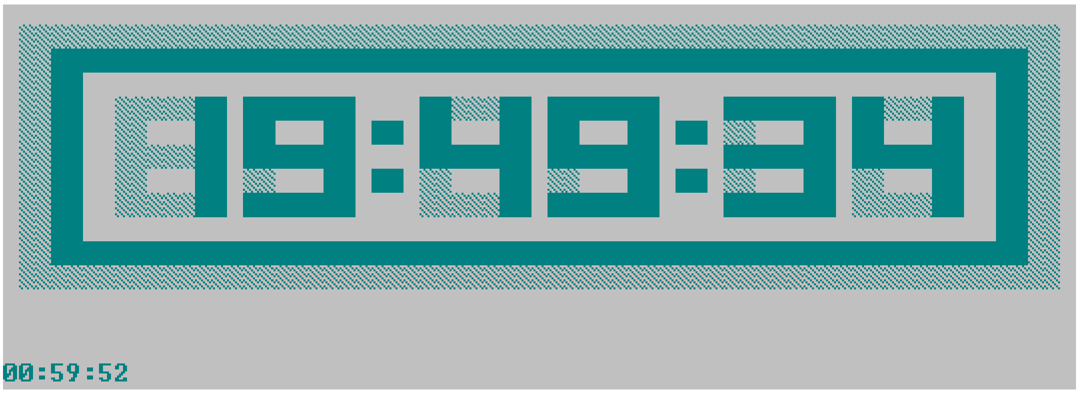
“d”, “m” and “y” are flags, i.e. you don’t need to respect the order of the example, all the combinations are possible: “dmy”, “dym”, “mdy”, “myd”, “ymd” and “ydm”.

## Clock

Use a clock.



```
>clock(1:0:0)
```



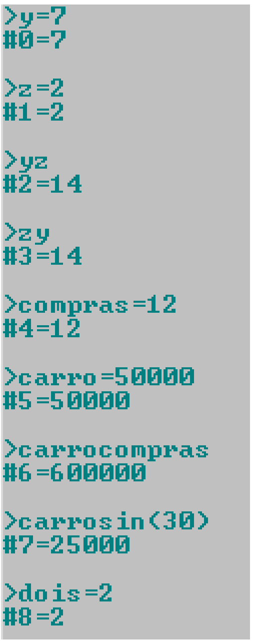
```
19:49:34



00:59:52
```

The example above shows a clock that will works during 1 hour.

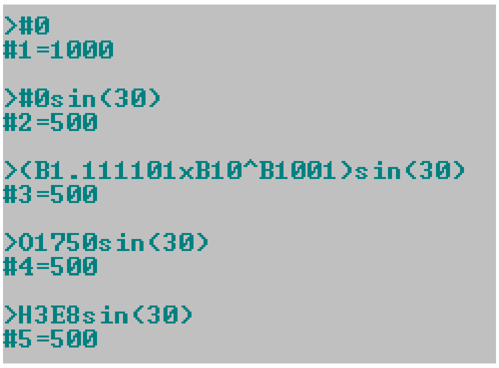
## Deduction of multiplications

Through smart algorithms Advanced Trigonometry Calculator provides capabilities to deduct multiplications in the expressions that you enter. So enter multiplication symbol is not completely needed in all cases of multiplications. Basically this calculator can detect different types of ways to enter values, and so it can deduct multiplications when between two values there's no arithmetical symbols.

Examples of this feature can be seen below:



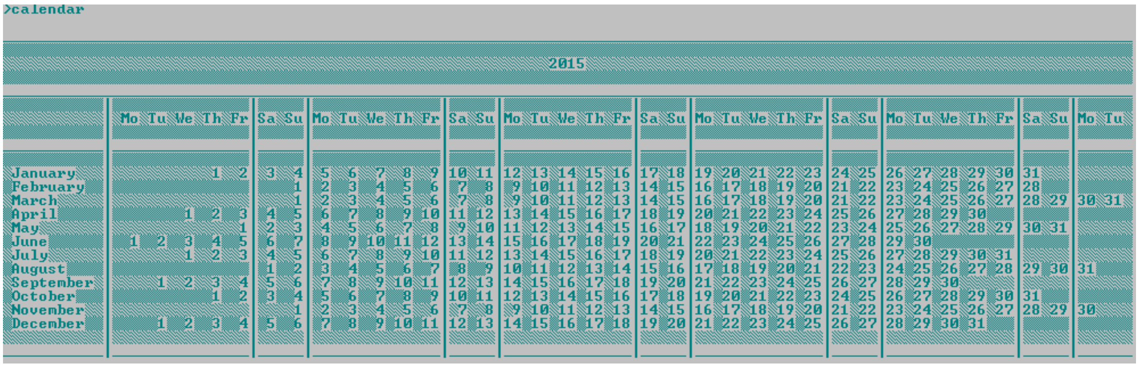
```
>y=7
#0=7

>z=2
#1=2

>yz
#2=14

>zy
#3=14

>compras=12
#4=12

>carro=50000
#5=50000

>carrocompras
#6=600000

>carrosin(30)
#7=25000

>dois=2
#8=2
```



```
>#0
#1=1000

>#0sin(30)
#2=500

>(B1.111101xB10^B1001)sin(30)
#3=500

>O1750sin(30)
#4=500

>H3E8sin(30)
#5=500
```

## Calendar

Check calendars.



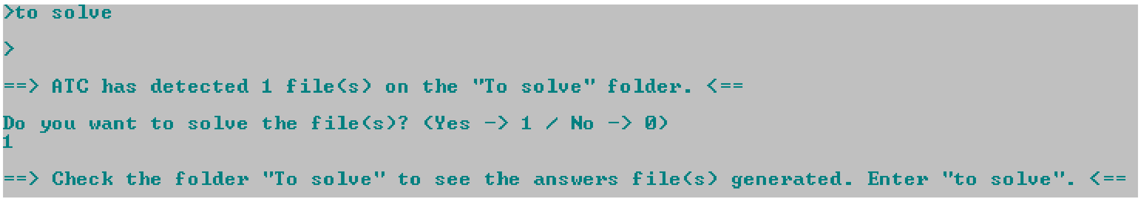
```
>calendar

                                                      2015

          Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu

January             1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
February                     1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
March                        1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
April           1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
May                   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
June       1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
July            1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
August                    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
September     1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
October             1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
November                     1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
December      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

This example shows the calendar that corresponds to the "current year". Entering "calendar(2016)" you will get the calendar for 2016.

## Txt detector

Txt detector provides a quick way to process text files (.txt). Txt detector detects ".txt" files that were not solved yet. It can demand the processing of multiple files at time. Its use is simple, just drag or create a ".txt" file on the "To solve" folder and run the app or if app it's already running press "Enter" button. You will be asked if you want to solve the detected files. You will find the folder "To solve" inside of the app folder, enter "to solve" to directly access the folder.

You can enable the feature in case you disabled it. Enter "enable txt detector".



```
>to solve

>

==> ATC has detected 1 file(s) on the "To solve" folder. <==

Do you want to solve the file(s)? (Yes -> 1 / No -> 0)
1

==> Check the folder "To solve" to see the answers file(s) generated. Enter "to solve". <==
```

# Download

Setup Advanced Trigonometry Calculator.exe – installation file.

Advanced Trigonometry Calculator.zip – portable file.

**Give your feedback contacting the developer!**

Via Facebook application page we can talk about it, give a “like” on the page. Click on 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 Electrotechnical and Telecommunications Engineering by Technology Higher School of Castelo Branco, part of Polytechnic Institute of Castelo Branco in Portugal.

Today Renato works at Altran Portugal in Lisbon, Portugal. As Junior Telecommunications Engineer.

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