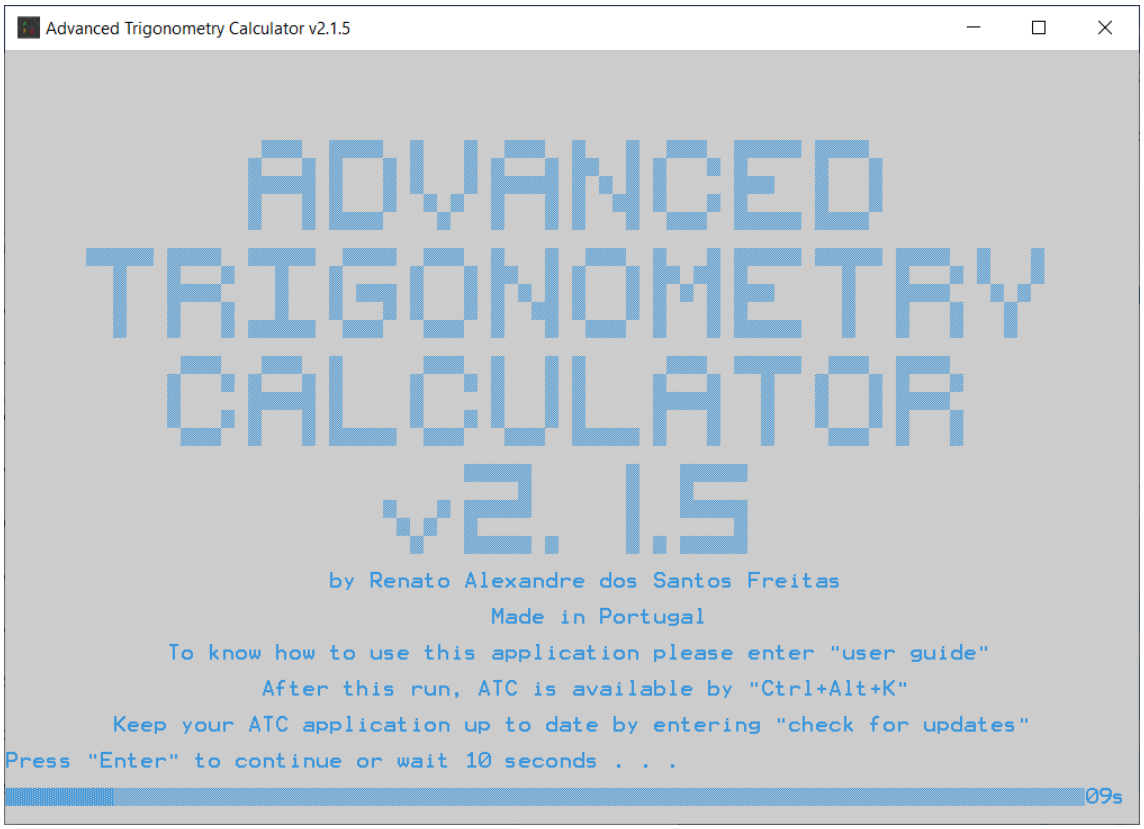
Advanced Trigonometry Calculator



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(), abs() 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".

e, pi, -INF and INF are mathematical values that user can make use to perform math calculations.

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.

Matrices

avg(), min(), max(), linsnum(), colsnum(), getlins(), getcols()

```
>var=1\2\3\4
#0=
1+0i 2+0i 3+0i 4+0i

>min(var)
#1=1

>max(var)
#2=4

>linsnum(var)
#3=1

>colsnum(var)
#4=4

>getcols(var\2\3)
#5=
3+0i 4+0i

>getlins(#5\0\0)
#6=
3+0i 4+0i

>getcols(#5\0\0)
#7=3

>matrix
#8=
0+0i 15+0i 30+0i 45+0i
60+0i 75+0i 90+0i 105+0i
120+0i 135+0i 150+0i 165+0i
180+0i 195+0i 210+0i 225+0i

>
```

```
>valores=0\30\45\60\90
#0=
0+0i 30+0i 45+0i 60+0i 90+0i

>sin(valores)
#1=
0+0i 0.5+0i 0.707107+0i 0.866025+0i 1+0i

>cos(valores)
#2=
1+0i 0.866025+0i 0.707107+0i 0.5+0i 0+0i

>tan(valores)
#3=
0+0i 0.57735+0i 1+0i 1.73205+0i INF+0i
```

```
>valores=1\2\3\4\5
#9=
1+0i 2+0i 3+0i 4+0i 5+0i


>avg(valores)
#10=3

>
```

Commands

Handling App environment

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.
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.
clean history	Let you clean the file “history.txt” with your entered expressions, respective answers and current time.
user guide	Let you open this user guide.
update	Let you download the latest version of the application.
update x64	Let you download the latest x64 version of the application.
update portable	Let you download the latest portable version of the application.
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 all now	The same as above but ATC will restart automatically.
reset settings	Let you delete only the files that were created due to configurations in the application made by the user.
reset settings now	The same as above but ATC will restart automatically.
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.
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.

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.
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 the abbreviations created until this moment.
enable txt detector	Let you enable again the txt detector feature. Check the section "Features" looking for "Txt detector".
eliminate strings	Let you eliminate all the strings created until this moment.
see strings	Let you see a list with the strings names. You can see the strings accessing "Strings" folder by enter "strings".
auto solve txt	Let your txt file be automatically solver after that in the last of the file be detected the flag "SOLVE_NOW". Check the section "Handling features" looking for "Solving txt files automatically".
atc from cmd	Let you use atc from Windows command-line (cmd.exe). Check the section "Handling features" looking for "Run ATC from Windows command-line".
atc over cmd	Let run atc over cmd, i.e. when you are in cmd and enter this command you will be able to start doing your mathematical tasks in cmd as you were running ATC directly.
donate	Let you be redirected to a Web page where you can donate any monetary value, to help the development of this app.
atc facebook	Let you be redirected to the facebook page of this app.
atc sourceforge	Let you be redirected to the page where this app is originally release. In this page you always find the latest version available for download.
current settings	Let you know the current state of the ATC settings. This way, you will be able to easily configure ATC for your customized use. Check the subsection "Handling features" to find an image illustrating what would be shown to you. Look for "Current settings".
auto adjust window	Let ATC auto adjust the ATC window to the current screen.
disable atc intro	Let you disable ATC intro window. This way you will go directly to the environment-resolution calculations.

enable atc intro	Let you enable ATC intro window. This way you will face intro window again with some info like the command to access this user guide.

Handling Calculations

Commands	Action
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.
verbose resolution	Let you enable or disable the feature of verbose resolution. Check “Calculations feature” subsection at the section “Features”.
see variables	Let you see the created variables with their values.
renamed variables	Let you see the created variables that have automatic renaming 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”.
see results	Let you see the calculated results with their values.
eliminate results	Let you eliminate all the calculated results 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” looking for “Equations systems solver”.
solve quadratic equation	Let you solve quadratic equation. Example of use: “solve quadratic equation(2\7\12)” resulting on the solutions “x1=-1.75+1.71391i” and “x2=-1.75-1.71391i”. Check the section “Features” looking for “Quadratic equations solver”.
solve equation	Let you know the zeros of a function (polynomial). E.g., “x^3+8x^2+x-42” -> “solve equation(1\8\1_42)” -> x1=-7; x2=2; x3=-3
triangles rectangles solver	Let you discover all the unknows of any triangle rectangle. Angles must be entered in degrees. In the sub-section “Calculation features” you can check a screenshot as example of use of this feature. Look for “Triangles Rectangles Solver”.

arithmetic matrix solver	Let you make various operations: sum/subtraction of matrices, multiply a matrix by a complex/real number and multiplication of matrices. In the sub-section "Calculation features" you can check a screenshot as example of use of this feature. Look for "Arithmetic Matrix Solver".
roots to polynomial	Let you get polynomial equation from its roots. Example of use: "roots to polynomial(2\7\12)" resulting on the polynomial "(1+0i)x^3+(-21+0i)x^2+(122+0i)x+(-168+0i)". Check the section "Features" looking for "Roots to Polynomial"
financial calculations	Let you calculate finance formulas. You enter the inputs and get the outputs.
graph settings	Let you configure your graph settings. (Xmin, Xmax, Xscale…)
graph	Let you visualize a graph of one or more functions of your interest. You can visualize multiple functions, e.g: graph(30sin(x)\15cos(x)), also after the visualization, ATC will ask you to know if you want see the graph data in a table view.
geometry calculations	Let you calculate areas and volumes. You enter the inputs and get the outputs. Check the section "Features" looking for "Geometry Calculations".
statistics calculations	Let you calculate many parameters related with statistics. Check the section "Features" looking for "Statistics Calculations".
physics calculations	Let you solve many physics formulas getting the value unknown of the formula that you want to solve. Check the section "Features" looking for "Physics Calculations".
unit conversions	Let you convert parameters of time, temperature, weight, length, area, volume and angle. You can find an image as example in the Calculation features section, looking for "Unit Conversions".
microeconomics calculations	Let you calculate Microeconomics formulas. You enter the inputs and get the outputs.
simplify polynomial	Let you get your polynomials simplified processing addition, subtraction, multiplication, division and parentheses in order to simplify very complex polynomials. Example of use: "simplify polynomial(((x-8)(x-9)(x-12))/(x-8)+((x-12)(x-20)(x-24))/(x-12))" resulting on the polynomial "((1+0i)x^2+(_14+0i)x+(_284.333+0i))" Check the section "Features" looking for "Simplify Polynomial"
function study	Let you visualize the study of a given function. Example of a valid command: "function study((1-x^2)/(x^2-4))". You can find some

	images as examples in the Calculation features section, looking for “Function Study”.
fft	Let you calculate the Fast Fourier Transform. You can find an image as example in the Calculation features section, looking for “FFT and IFFT”.
ifft	Let you calculate the Inverse Fast Fourier Transform. You can find a image as example in the Calculation features section, looking for “FFT and IFFT”.
higher precision	Let enable ATC to provide values in optimized scientific notation, with at most 15 decimal places.

Handling App folders

Commands	Action
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”.
scripts examples	Let you open the folder of scripts examples.
user functions	Let you open the folder of user functions.
strings	Let you open the folder with the strings created by the user.

Handling Time

Commands	Action
day of week	Let you check the day of week corresponding to a certain date, e.g. “dayofweek(d11m7y2014)” has as response “Friday”.
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” looking for “Stopwatch”.
run stopwatch	The same as above but will have an independent window from atc for the stopwatch feature. E.g “run stopwatch(3)”.
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” looking for ”Timer”.
run timer	The same as above but will have an independent window from atc for the timer feature. E.g. "run timer(0:5:0)".
big timer	Let you control the time you spend for anything. Entering “big timer(0:5:0)” you will be notified when passed 5 minutes since the press of button “Enter”. Check the section “Features” looking for “Big timer”.
run big timer	The same as above but will have an independent window from atc for the timer feature. E.g. "run big timer(0:5:0)".

clock	Let you use a clock. Entering "clock(0:5:0)" you will have a clock during 5 minutes. Check the section “Features” looking for “Clock”.
run clock	The same as above but will have an independent window from atc for the clock feature. E.g. "run clock(0:5:0)".
big clock	Let you use a big clock. Entering "big clock(0:5:0)" you will have a clock during 5 minutes. Check the section “Features” looking for “Big clock”.
run big clock	The same as above but will have an independent window from atc for the clock feature. E.g. "run big clock(0:5:0)".
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)”.
actual time response	Let you enable or disable the functionality of show the current time after having been shown an answer for an expression entered.
time difference calculations	Let you know the difference time between two dates with time.

Handling PC

Commands	Action
shutdown	Let you shut down your PC.
shutdown now	Let you shut down your PC immediately.
restart pc	Let you restart your PC.
restart pc now	Let you restart your PC immediately.
hibernate	Let you hibernate your PC. Administrator privileges are required.
log off	Let you log off your PC.
sleep	Let you sleep your PC. Administrator privileges are required.
lock	Let you lock your PC.

Sorting

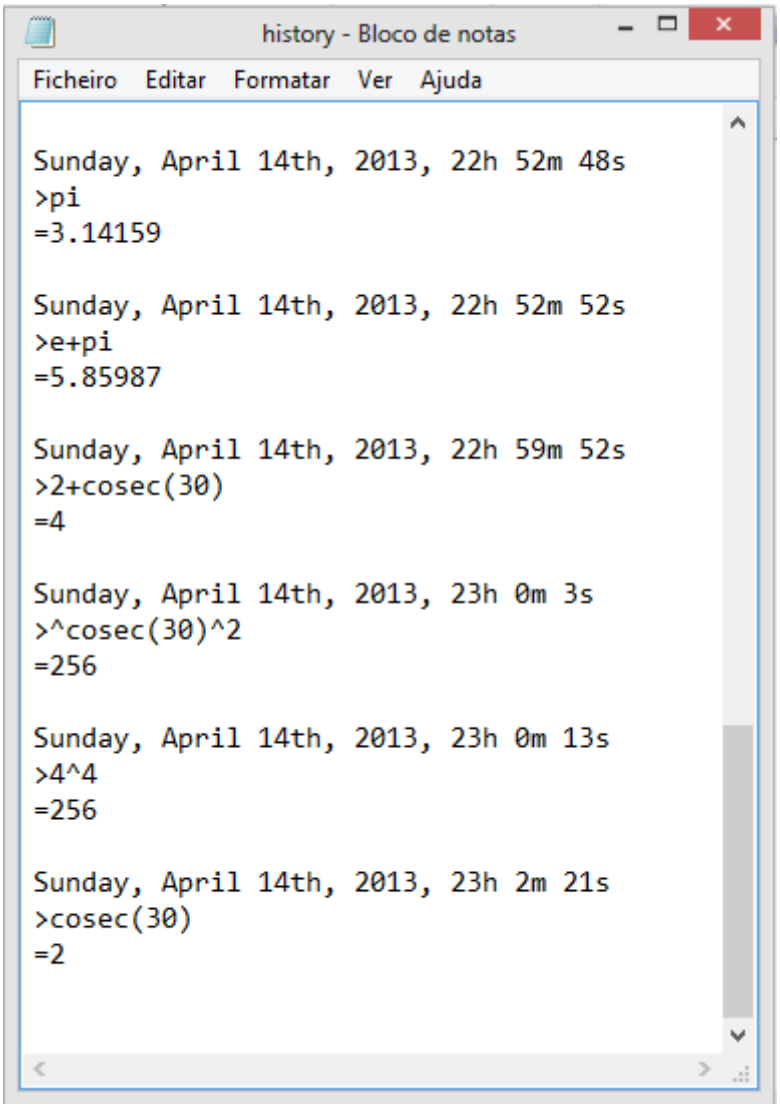
Commands	Action
ascending order	Let you put a set of real values in ascending order. Example: Input: ascending order(_1\5_12\14_67\50) Output: -67, -12, -1, 5, 14, 50
descending order	Let you put a set of real values in descending order. Example: Input: descending order(_1\5_12\14_67\50) Output: 50, 14, 5, -1, -12, -67
ascii order	Let you put a set of real values in ascii order. Check the section of Sorting
inverse ascii order	Let you put a set of real values in inverse ascii order. Check the section of Sorting

Features

Handling features

History

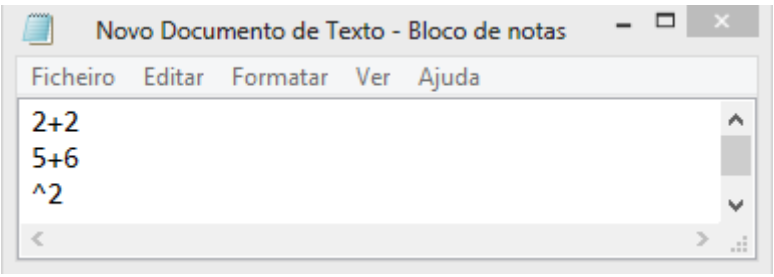
To help you this application creates a text file with the expressions that you have entered, their answers and current time. Its name is "history.txt" you can find it inside the folder that contains the application executable in use. You can also enter “history” and see the file automatically. So, an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



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:

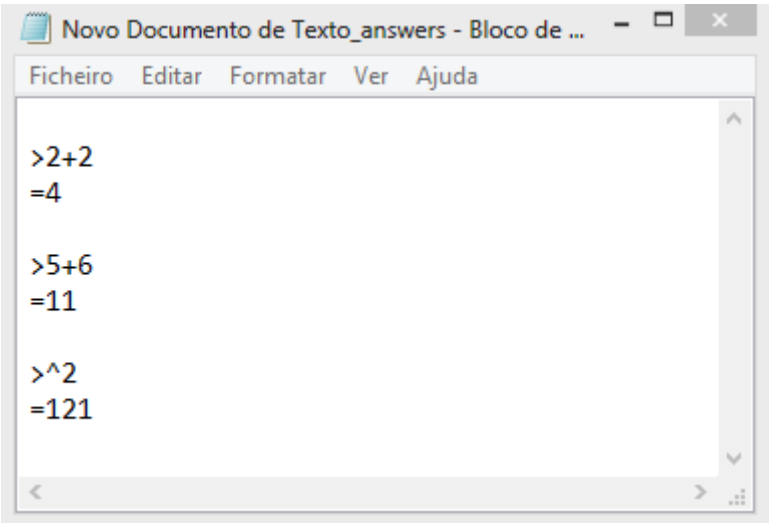
Original file:



Interaction with application:

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

New file with answers:



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".

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.

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.

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 is 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". <==
```

Solving txt files automatically

You can automatically solve the txt file that you are editing for ATC.

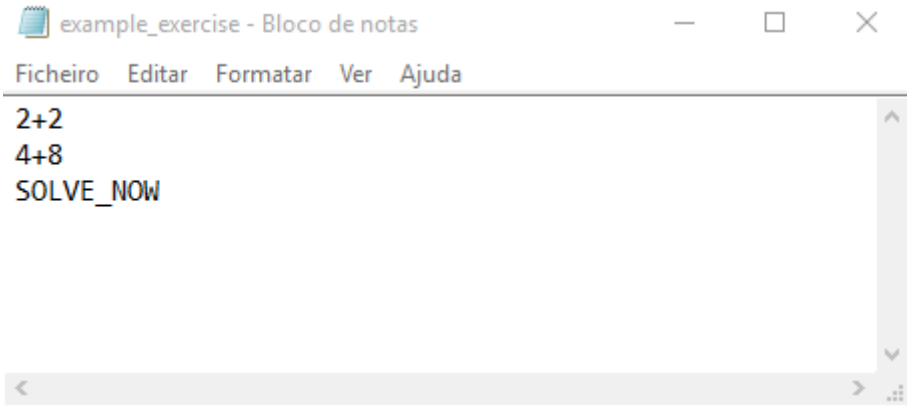
1) Access ATC and enter: "auto solve txt"

```
>auto solve txt
==> Drag to here the file for automatic processing and press the button "Enter" <==
```

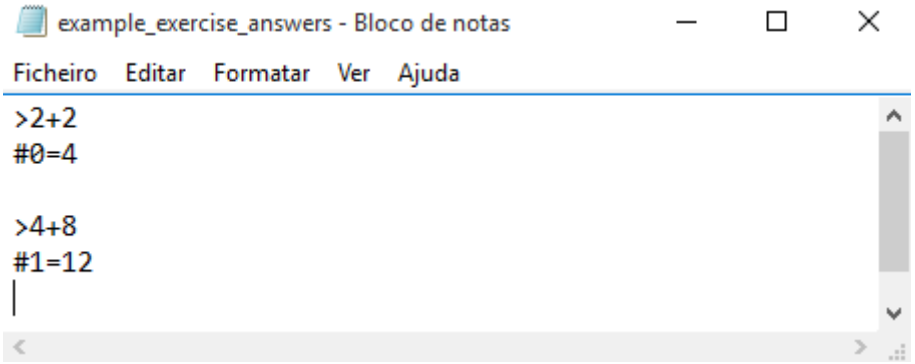
2) Drag the txt file that you are editing to the environment-resolution calculations of ATC and press “Enter”.

```
>auto solve txt
==> Drag to here the file for automatic processing and press the button "Enter" <==
C:\Users\Renato\Desktop\example_exercise.txt
==> Waiting for the flag "SOLVE_NOW" be detected in the last line of this file. <==
```

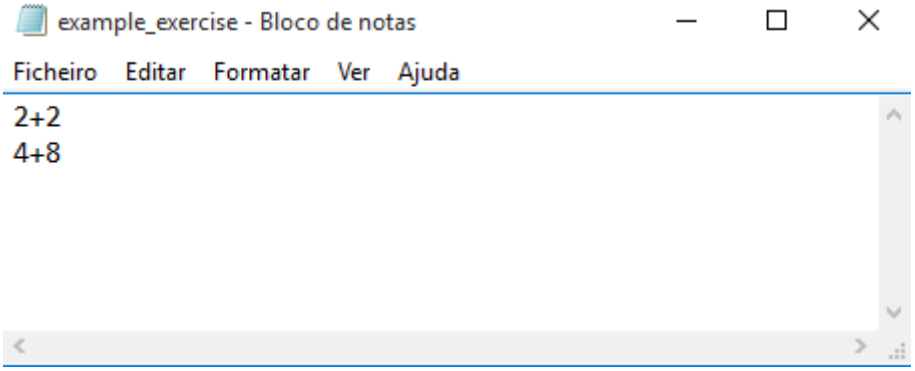
3) Edit your file. You must not forget to put the flag in the last line when your file is ready to be solved.



4) And to automatically solve it: Press “Ctrl+s” (e.g., in case you are using notepad.exe). The file with the answers must appear in instants.



5) Note that the flag “SOLVE_NOW” is automatically removed from the original file after it has been processed.



Run ATC from Windows command-line

Have you ever dreamed with a great calculator for Windows command-line?

All you need to do is enter the command “atc from cmd” and permit changes giving administrator privileges to perform the activation of the feature.

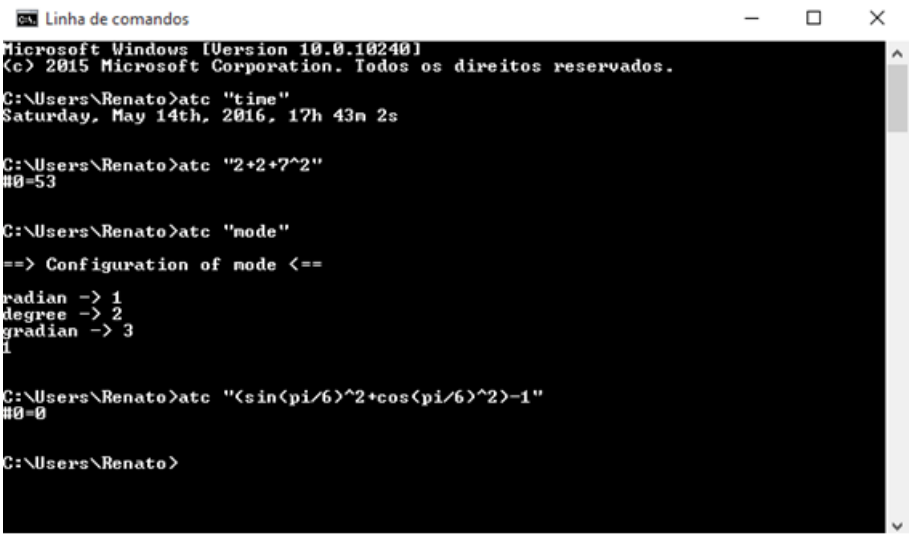
Activation steps:

1) Enter the command “atc from cmd” in ATC.

```
>atc from cmd_
```

2) You will be asked to permit changes by Windows command-line. You need to click on “Yes” button.

3) If you have said yes to the changes, you can now, try use atc from cmd. The syntax to use is simple as shown below. Note that you need to open a new Windows command-line after the activation.



4) Now use your imagination and enjoy Advanced Trigonometry Calculator in your Windows command-line.
5) Now with the activation you can easily open atc. Just open cmd.exe and enter “atc”.

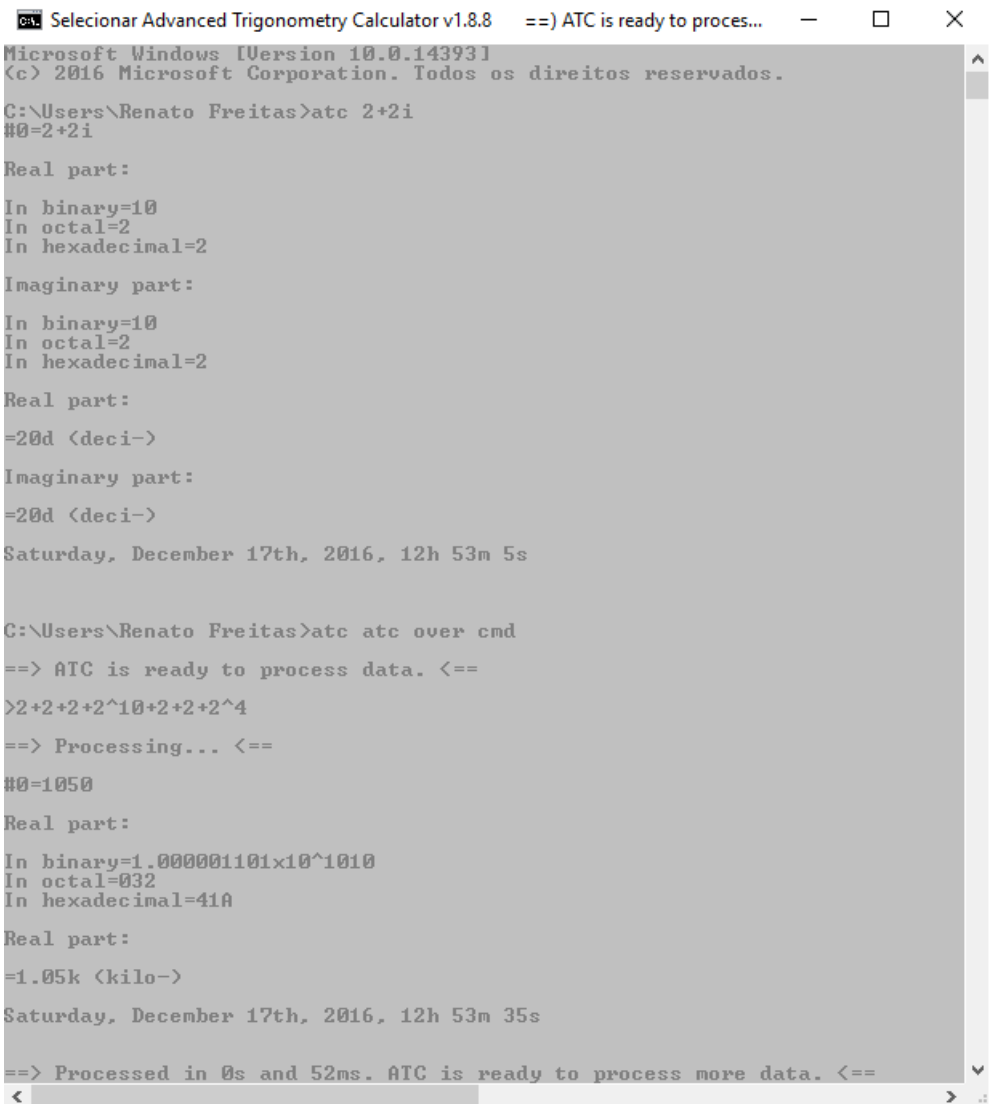
This feature permits the complete use of ATC in any user path where user are navigating in the Windows command-line.

Note: If for some reason you want disable this feature. Check this link: http://www.howtogeek.com/118594/how-to-edit-your-system-path-for-easy-command-line-access/

Run ATC over Windows command-line

After entering the command “atc over cmd” you will be able to work in cmd as if you were running ATC directly by double click on the ATC executable. For reproduce this example, please don’t forget entering the command “atc from cmd” and activate that feature.

Below is an example about what you can do.



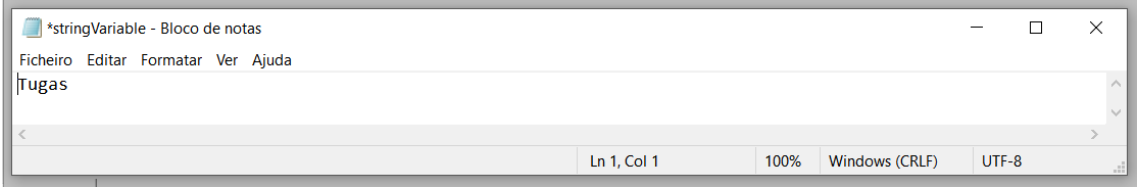
Current settings

See the current settings of ATC to easily manage its use.

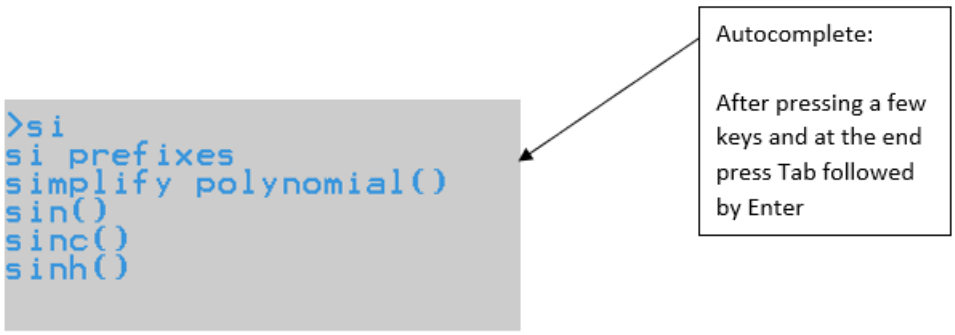
```
>current settings

Mode-------------------------------------> Radian DEGREE Gradian ¦ Info: Enter "mode" to change.
Numerical Systems Response---------------> Enabled DISABLED ¦ Info: Enter "numerical systems" to change.
SI Prefixes Response---------------------> ENABLED Disabled ¦ Info: Enter "si prefixes" to change.
Actual Time Response---------------------> Enabled DISABLED ¦ Info: Enter "actual time response" to change.
Verbose Resolution-----------------------> Enabled DISABLED ¦ Info: Enter "verbose resolution" to change.
Colors-----------------------------------> Text: AQUA - Background: WHITE ¦ Info: Enter "colors" to change.
Window-----------------------------------> x: 0 - y: 0 - Width: 1000 - Height: 1000 ¦ Info: Enter "window" to change.
Dimensions-------------------------------> Lines: 150 - Columns: 150 ¦ Info: Enter "dimensions" to change.

>
```

Opentxt

```
>Tugas="A 100%!"

>opentxt("C:\Users\renat\Pictures\Advanced Trigonometry Calculator\stringVariable.txt")
==> Close the file that was open to continue. <==
```



Autocomplete



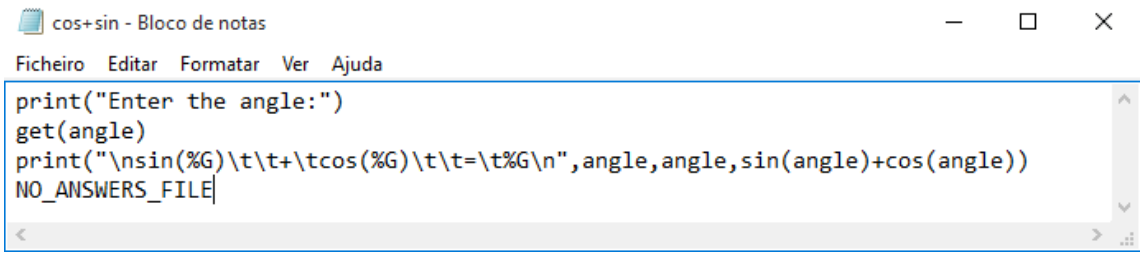
Scripting features

print, get, and sprint

The goal of the scripting feature is to provide quick means to perform tasks repeatedly.

The scripting feature comprises the use of all mathematical functions, commands and features that the user can enjoy in the “environment-resolution calculations” and it is an extension of the processing of text files (.txt) feature. ATC folder has a sub folder called “Scripts examples” that you can easily access by enter “scripts examples”. As the folder name says it is a folder with examples of scripts that you can use/modify, and they serve as example for you to create your own atc scritps.

Script example:



Its output:

```
Enter the angle:
30

sin(30)         +       cos(30)         =       1.36603
```

You can find the script example above in the folder “Scripts examples”. Just drag and drop the file “cos+sin.txt” to the ATC environment-resolution calculations and press “Enter” key you will face the output.

As you may already understood there are print() and get() functions to print the output and get the input, respectively.

The print() function is a emulation of the printf() from C programming language, so the syntax is truly the same. If you know how to use printf() from C you know how to use print() function.

"NO_ANSWERS_FILE" is a flag that informs the app that you doesn't want save and see the answers file that is created after processing the inputted txt file .

```
>get(hoje)
23

>hoje
#5=23

>get(maravilha)
"Portugal"

>print(maravilha)
Portugal
```

The get() function can read a string or an expression:

- To save a string you need the quotation marks for atc understand the input as text and save the string.
    - Strings are saved in the folder "Strings" that is accessible by enter "strings" and can be listed by enter "see strings".
- To save a value all that you need is to respond with an expression, atc will save the solution value as a variable.

```
>sprint(dateAndTime,"%d-%d-%d %d:%d",24,2,2016,22,14)

>print(dateAndTime)
24-2-2016 22:14


>print("%s",dateAndTime)
24-2-2016 22:14
```

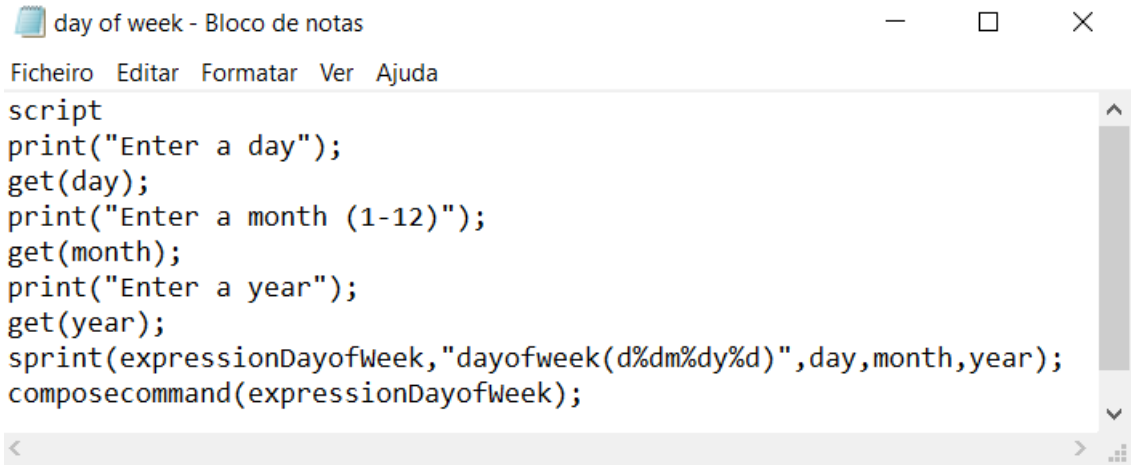
The sprint() function is a emulation of the sprintf() from C programming language, such as print() from printf().

As you may observe above, you have two ways to use the print(): you can print a string, in this case "dateAndTime" as print() acts like puts() from C programming language or you can use the printf() text syntax.

You can easily check the printf() syntax from the url:

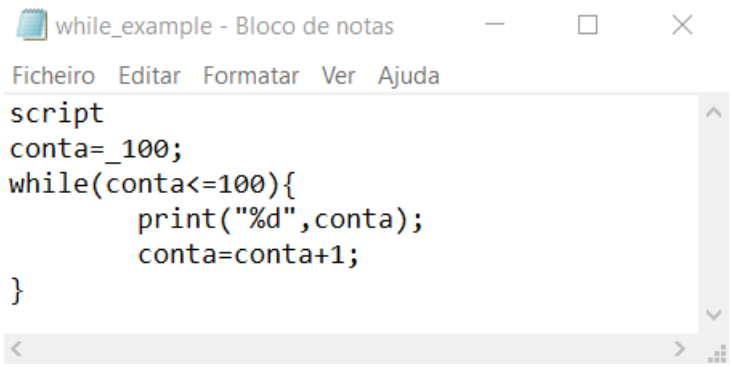
http://www.cplusplus.com/reference/cstdio/printf/

Compose commands



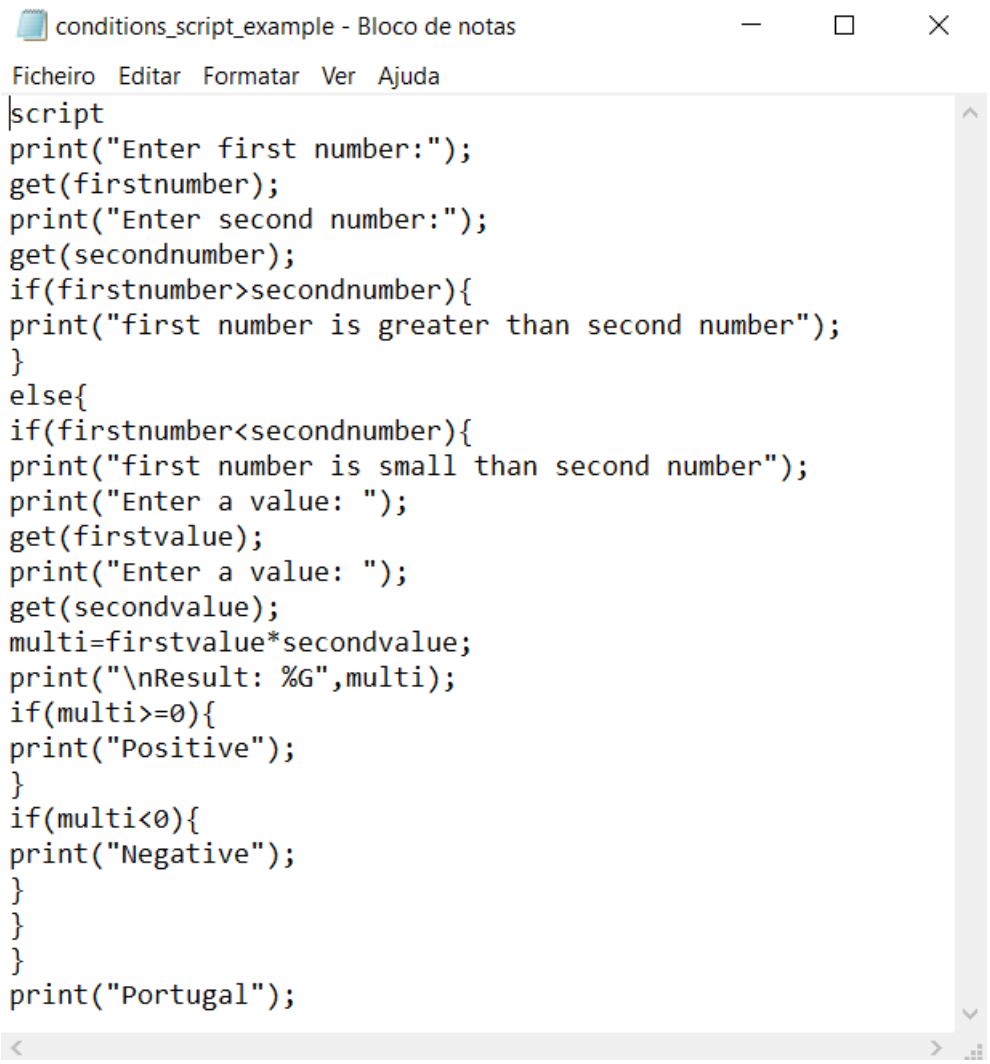

```
Enter a day
7
Enter a month (1-12)
4
Enter a year
2018

y2018m4d7=Saturday
>
```

while and for

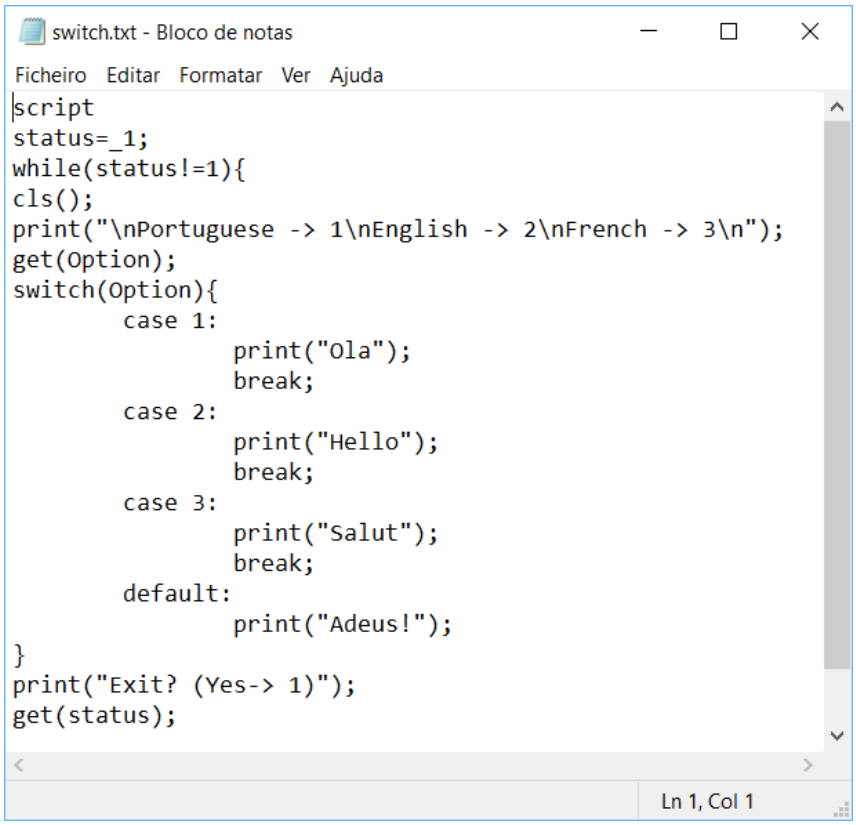



```
script

print("\n\nMultiplication Table 1-100\n");

x=0;

j=0;

for(x=1;x<=100;x=x+5){

        for(j=1;j<=10;j=j+1){


        print("%dx%d=%d\t%dx%d=%d\t%dx%d=%d\t%dx%d=%d\t%dx%d=%d",x,j,x*j,x+1,j,(x+1)*j,x+2,j,(x+2)*j,x+3,j,(x+3)*j,x+4,j,(x+4)*j);

        }

        print(" ");

}
```

if and else

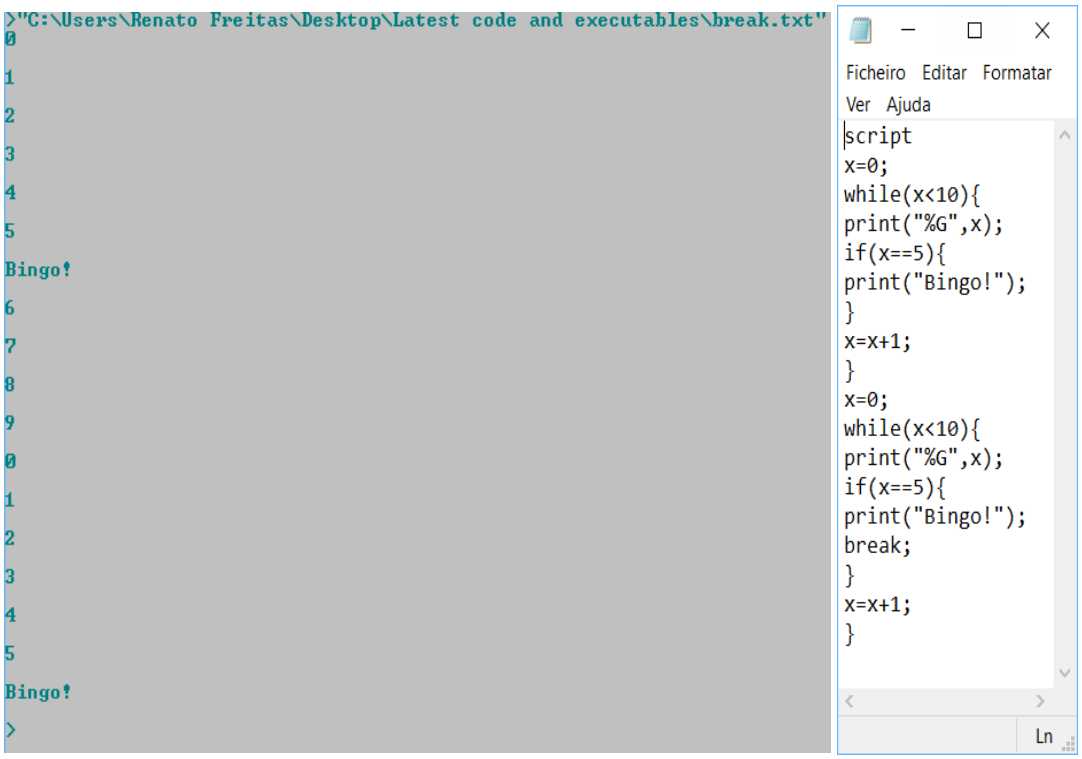

conditions_script_example - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
script
print("Enter first number:");
get(firstnumber);
print("Enter second number:");
get(secondnumber);
if(firstnumber>secondnumber){
print("first number is greater than second number");
}
else{
if(firstnumber<secondnumber){
print("first number is small than second number");
print("Enter a value: ");
get(firstvalue);
print("Enter a value: ");
get(secondvalue);
multi=firstvalue*secondvalue;
print("\nResult: %G",multi);
if(multi>=0){
print("Positive");
}
if(multi<0){
print("Negative");
}
}
}
print("Portugal");

Switch case and cls()

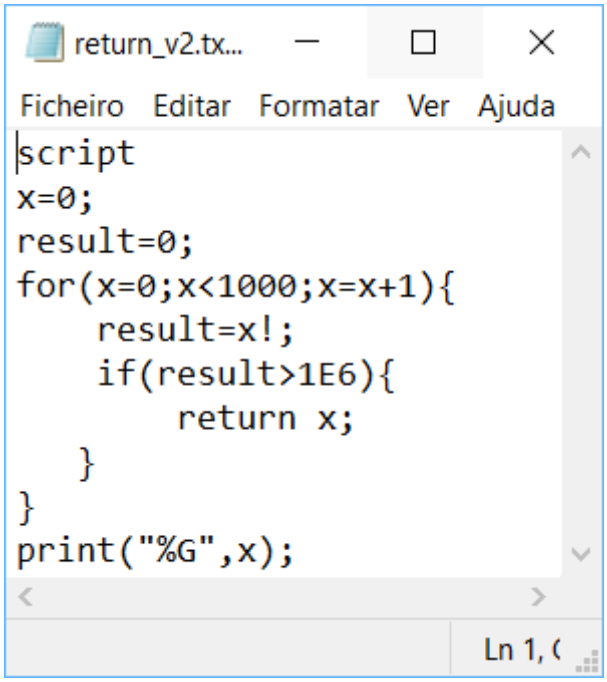



```
Portuguese -> 1
English -> 2
French -> 3


1

Ola

Exit? (Yes-> 1)
```

Break



Return



```
>"C:\Users\Renato Freitas\Desktop\Latest code and executables\return_v2.txt"
11
>
```

Replace

```
script
text="AAAAA";
char="A";
charnew="B";
```

```
>"C:\Users\renat\Pictures\Advanced Trigonometry Calculator\Scripts examples\replace.txt"



BBBBB
```

Change -> text="CAAAC";

```
>"C:\Users\renat\Pictures\Advanced Trigonometry Calculator\Scripts examples\replace.txt"



CBBBC
```

Syntax of replace function:

replace(data_to_find\replacement_to_found_data\expression_with_data\0_or_1_(meaning_below))

0 means all found cases

1 means just the first found case

Simple way to memorize: 1/0 is equals to infinity and 1/1 is equal to 1.

Replace By Index

```
script
text="AAAAA";
char="A";
charnew="B";
newtext=replacebyindex(char\charnew\text\2\0);
print("%s",newtext);
```

```
>"C:\Users\renat\Pictures\Advanced Trigonometry Calculator\Scripts examples\replaceByIndex.txt"


AABBB
```

Syntax of replacebyindex function:

replacebyindex(data_to_find\replacement_to_found_data\expression_with_data\index_start_replacing\0_or_1_(meaning_below))

0 means all found cases

1 means just the first found case

Simple way to memorize: 1/0 is equals to infinity and 1/1 is equal to 1.

The index_start_replacing is the value of the real text index minus 1, for scripting all indexes start on 0.

Count Occurrences

```
script
print("Enter a expression:");
get(totaldata);
print("Enter the data to count the occurrences in the expression:);
get(data);
value=countoccurrences(data\totaldata);
print("Data has been found %d times in the expression.",value);
```

```
>"C:\Users\renat\Pictures\Advanced Trigonometry Calculator\Scripts examples\countOcurrences.txt"
Enter a expression:
"AAAAAAAAAAAAAAAAAAAAAAAAAA"

Enter the data to count the occurrences in the expression:
"A"

Data has been found 26 times in the expression.
```

Delete X Occurrences

```
script

print("Enter a expression:");

get(TotalTxt);

print("Enter the data to be deleted:);

get(Txt);

print("Enter the number of times to be deleted:");

get(num);

Text=deletexoccurrences(Txt\TotalTxt\num);

print("Result: %s",Text);
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\deleteXoccurrences.txt"
Enter a expression:
"AAAAAAAAAAAAAAAAAAABBBB"

Enter the data to be deleted:
"B"

Enter the number of times to be deleted:
4

Result: AAAAAAAAAAAAAAAAA
>
```

Is Contained

```
script

print("Enter a expression:");

get(totalData);

print("Enter a bit of the expression:");

get(data);

if(iscontained(data\totalData)){

        print("As expected! Thank you!");

}

else{

        print("Oh no... you have forgotten the bit.);

}
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isContained.txt"
Enter a expression:
"Eu sou português!"


Enter a bit of the expression:
"Eu sou"


As expected! Thank you!
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isContained.txt"
Enter a expression:
"Eu sou português!"


Enter a bit of the expression:
"Eu não sou"


Oh no... you have forgotten the bit.
>
```

Calc

```
script

continue=1;

while(continue==1){

print("Enter a expression:");

get(data);

Result=calc(data);

print("Result: %G",Result);

print("Continue? (Yes-> 1/No-> any value)");

get(continue);

}
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\calc.txt"
Enter a expression:
"2+2+2"

2+2+2
Result: 6
Continue? (Yes-> 1/No-> any value)
0
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\calc.txt"
Enter a expression:
"tan(30)/tan(30)"

Result: 1
Continue? (Yes-> 1/No-> any value)
1
Enter a expression:
"2+2+2+2+2+2+2+2+2+22+2"

Result: 42
Continue? (Yes-> 1/No-> any value)
0
>
```

Is Equal

```
script
print("Enter a expression:");
get(totalData);
print("Enter again the same expression:");
get(data);
if(isequal(data\totalData)){
        print("As expected! Thank you!");
}
else{
        print("Oh no... the expressions don't match.);
}
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isEqual.txt"
Enter a expression:
"Water for me"

Enter again the same expression:
"Water for me"

As expected! Thank you!
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isEqual.txt"
Enter a expression:
"Water for me"

Enter again the same expression:
"Water for you"

Oh no... the expressions don't match.
>
```

Get Positive Value

```
script

print("The expected is that the value will not be accepted.");

print("Enter a value negative:");
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\getPosValue.txt"
The expected is that the value will not be accepted.

Enter a value negative:

-8

Error: Value must be positive.

8

Value: 8
```

Is To Write

```
script

print("First case:");

data="NO_ANSWERS_FILE";

print("%s",data);

if(istowrite(data)){

        print("ATC will create a file with answers.");

}

else{

        print("ATC will not create a file with answers.");

}

print("Second case:");

Data="ANSWERS_FILE";

print("%s",Data);

if(istowrite(Data)){

        print("ATC will create a file with answers.");

}

else{

        print("ATC will not create a file with answers.");

}
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isToWrite.txt"
First case:

NO_ANSWERS_FILE
ATC will not create a file with answers.
Second case:

ANSWERS_FILE
ATC will create a file with answers.
>
```

Is Variable

```
script

script

print("First case is valid variable");

data=12;

data="data";

if(isvariable(data)){

        print("Great! That is a variable!");

}

else{

        print("Ups... not a variable.");

}

print("Second case is an invalid variable");

if(isvariable(daTa)){

        print("Great! That is a variable!");

}

else{

        print("Ups... not a variable.");

}
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isVariable.txt"
First case is valid variable

Great! That is a variable!

Second case is an invalid variable

Ups... not a variable.

>
```

Is Contained Variable

```
script

print("First case is a present valid variable");

value=12;

value="value"

data="value+7";

if(iscontainedvariable(value\data)){

	print("Great! Your variable has been found!");

}

else{

	print("Ups... variable not found.");

}

print("Second case is an not present variable");

value=12;

data="sqrt(7)";

if(iscontainedvariable(value\data)){

	print("Great! Your variable has been found!");

}

else{

	print("Ups... variable not found.");

}
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isContainedVariable.txt"
First case is a present valid variable


Great! Your variable has been found!

Second case is an not present variable


Ups... variable not found.

>
```

Is Contained By Index

```
script

print("Enter a expression:");

get(totalData);

print("Enter a bit of the expression:");

get(data);

print("Enter the index:");

get(index);

index="index"

if(iscontainedbyindex(data\totalData\index)){

	print("As expected! Thank you!");

}

else{

	print("Oh no... you have forgotten the bit or provide wrong index number.);

}
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isContainedByIndex.txt"
Enter a expression:

"AAAAAABBBB"

Enter a bit of the expression:

"A"

Enter the index:

8

Oh no... you have forgotten the bit or provide wrong index number.

>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\isContainedByIndex.txt"
Enter a expression:

"AAAAAABBBB"

Enter a bit of the expression:

"A"

Enter the index:

2

As expected! Thank you!

>
```

Strlen

```
script

print("Enter a expression:");

get(data);

size=strlen(data);

print("Length of expression: %G",size);
```

```
>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\strlen.txt"
Enter a expression:

"Ola ola"


Length of expression: 7

>"C:\Users\renat\Desktop\ATC Projects\Advanced Trigonometry Calculator\Advanced Trigonometry Calculator\Scripts examples\strlen.txt"
Enter a expression:

"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"


Length of expression: 142

>
```

Calculation 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
```

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

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
```

Equations systems solver

You can solve equations systems. Take an example:

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

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

```
>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.

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

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

```
>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
```

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 to create expressions easier than with just numeric digits and functions.

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
```

Numerical systems places

You can choose how many decimals, 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 to 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=1.10001000101100101111x10^1011
In octal=6105.4560507534
In hexadecimal=C45.970A3D70A3F330002023D

>hp10hppi*1000
#3=3141.59
In binary=1.10001000101100101111x10^1011
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.
```

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 and answer

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, 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
```

Also, instead of entering the previous result as value you can enter “ans”. It’s useful when you want to do more calculations related with the last answer.

```
>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
```

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
```

User functions

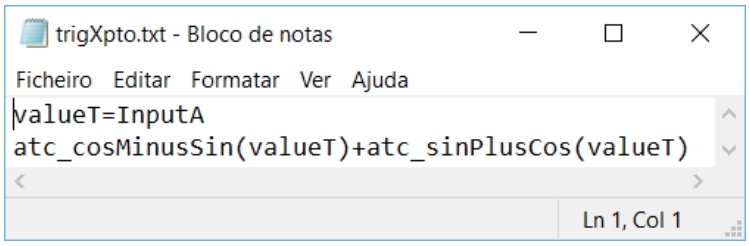
Through scripting feature, ATC is able to process user functions. A user function is a script that has a variable called “InputX” X is replaced by A, B, C, D as the number of inputs increase and on what the script last line with a mathematical expression will have its solution as the returned value of the user function.

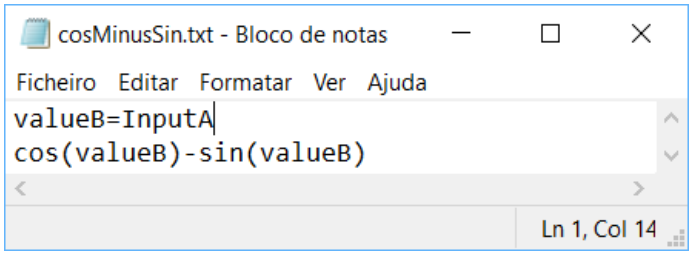
You can easily convert your script into a user function changing all the file name characters to Latin alphabet letters (uppercase and lowercase).

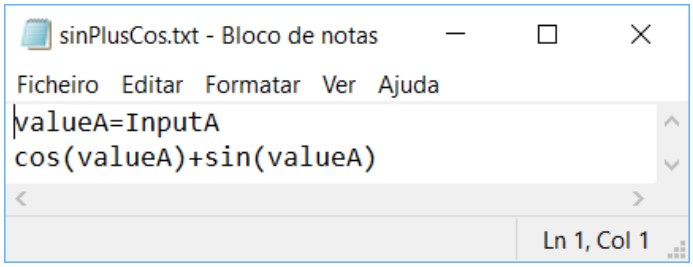
User functions are accessible by the command “user functions”

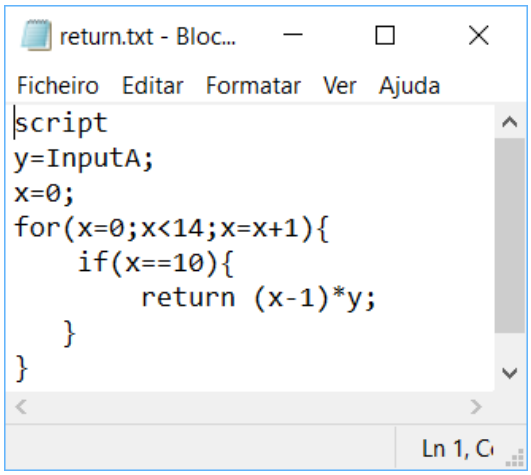
```
>user functions

>atc_trigXpto(30)
#0=1.73205

>atc_trigXpto(30)+atc_trigXpto(30)
#1=3.4641
```

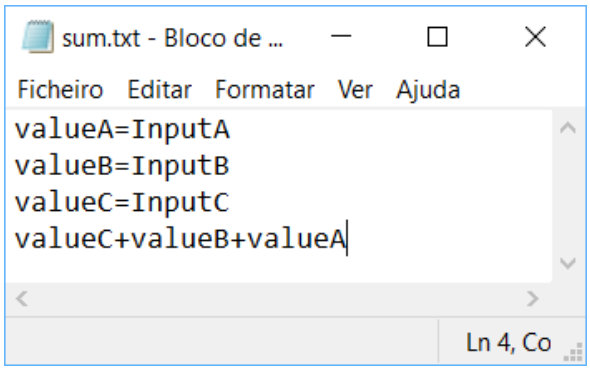
```
>atc_trigXpto(30)
#6=1.73205

>atc_trigXpto(30)+atc_trigXpto(30)
#7=3.4641

>atc_return(30)
#0=270

>atc_sum(4\5\6)
#1=15

>atc_multiply(5)
#2=5+5i
```







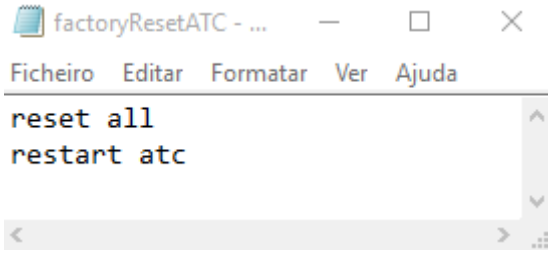




After entering "user functions" will be opened the folder where you must include your user functions to them be treated as user functions, and so you can enjoy your own functions inside your mathematical expressions.

User functions are a new world for ATC users. You will find it a feature very useful.

Factory reset ATC with a user function!




```
>atc_factoryResetATC()
```

Quadratic equations solver

ATC can solve quadratic equations by applying the quadratic formula (below).

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

An example to solve:

$$2x^2 + 7x + 4 = 0$$

Identify a, b and c:

a=2 b=7 c=4

Now can solve:

```
>solve quadratic equation(2\7\4)

x1=-0.719224
x2=-2.78078
```

Note that a, b and c could be complex numbers!

Take an example:

$$(2 + 8\mathrm{i})x^2 + (3 - 5\mathrm{i})x + 10 - 4\mathrm{i} = 0$$

```
>solve quadratic equation(2+8i\3-5i\10-4i)

x1=-0.646247-0.5417i
x2=1.14625+1.0417i
```

Verbose resolution

Verbose resolution is a feature that allows the user to know how its math expression has been processed until the result, the answer, have been obtained by ATC.

This feature helps users to learn and explore math resolution of math expressions.

Function domain error

```
>sin(0/0)
Error in function domain.
 ==> For sine function the valid domain is [-INF,INF].
 ==> Your function argument: -NAN(IND)
```

Solver

```
>solver(x-(e+pii))
#0=2.71828+3.14159i
```

```
>solver(sqrt(x)-10)
#2=100
```

```
>solver(x-(2+8i))+solver(x-(pi+ei))
#1=5.14159+10.7183i
```

```
>solver(sqrt(x)-12)
#0=144

>solver(1/(x-8)+2x-4)
#1=2.08452

>x=#1
#2=2.08452

>1/(x-8)+2x
#3=3.99999

>_
```

```
>solver(sin(x)-sin(7+4i))
#0=7+4i

>solver(sin(x)-sin(7+4i)+cos(14+8i)-cos(2x))
#1=7+4i
```

```
>solver(cos(x)+sin(2x)+tan(x)-tan(30)-sin(60)-cos(30))
#4=30
```

```
>solver(sin(x)-cos(30))
#5=60

>solver(sin(x)-cos(45))
#6=45

>solver(sin(x)-cos(90))
#7=0

>solver(sin(x)-cos(60))
#8=30
```

Definite Integral

$$\int_0^{\pi} \sin(x) = 2$$

```
>mode

==> Configuration of mode <==

radian -> 1
degree -> 2
gradian -> 3
1

>solver(0\pi\sin(x))
#1=2.00033

>
```

Equation Solver

ATC can solve polynomials.

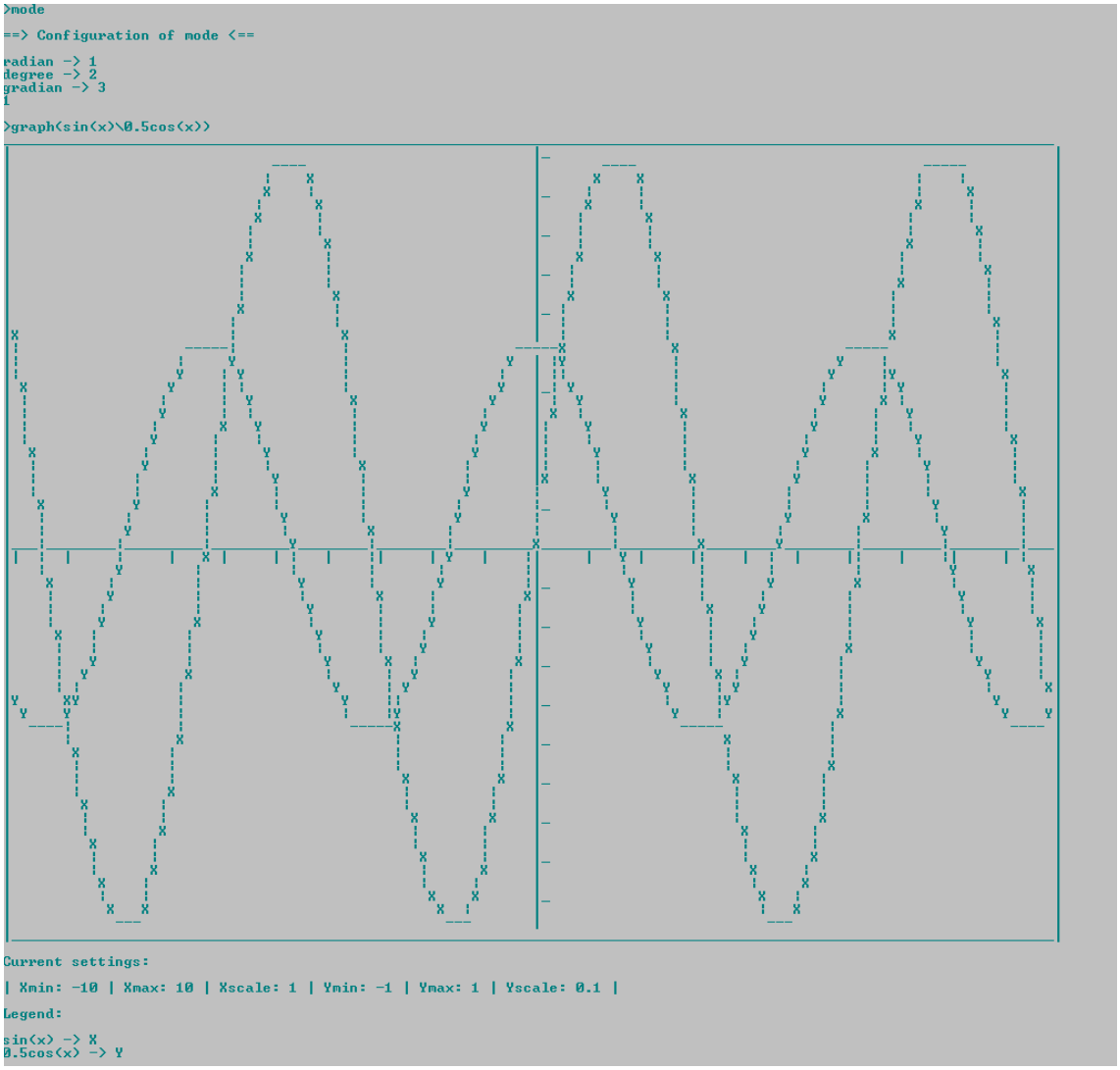
```
>roots to polynomial(2+5i\2-5i\7+2i\7-2i\pi+ei\pi-ei)

(1+0i)x^2+(-4+0i)x+(29+0i)
(1+0i)x^3+(-11+-2i)x^2+(57+8i)x+(-203+-58i)
(1+0i)x^4+(-18+0i)x^3+(138+0i)x^2+(-618+0i)x+(1537+0i)
(1+0i)x^5+(-21.1416+-2.71828i)x^4+(194.549+48.9291i)x^3+(-1051.54+-375.123i)x^2+(3478.5+1679.9i)x+(-4828.63+-4178i)

Final polynomial:
(1+0i)x^6+(-24.2832+0i)x^5+(268.356+0i)x^4+(-1795.74+-5.68434E-14i)x^3+(7801.7+-2.27374E-13i)x^2+(-20323.1+0i)x+(26526.6+0i)

Export result? (Yes -> 1 \ No -> 0)
0

>solve equation(1\_24.2832\268.356\_1795.74\7801.7\_20323.1\26526.6)

x1=6.99991+1.99957i
x2=6.99991-1.99957i
x3=3.14161+2.71843i
x4=2.00007+4.99998i
x5=2.00007-4.99998i
x6=3.14161-2.71843i

Export result? (Yes -> 1 \ No -> 0)
0

>
```

```
>solve equation(1\44\656\3098\_10861\_124126\_181316\444120\856800)

x1=-17
x2=-12
x3=-10
x4=-2
x5=-7
x6=5
x7=2
x8=-3

Export result? (Yes -> 1 \ No -> 0)
0

>_
```

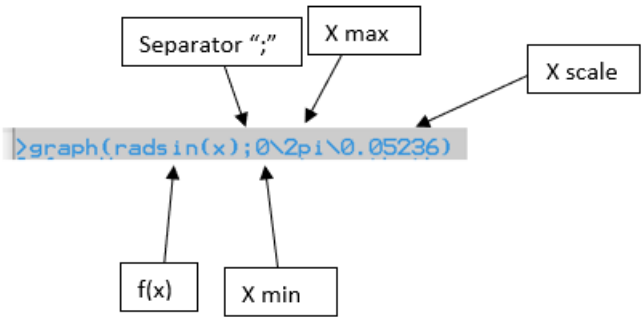
```
>solve equation(2x^2-7x-12)
x1=4.7604
x2=-1.2604

Export result? (Yes -> 1 \ No -> 0)
0

>solve equation(2\_7\_12)

x1=4.7604
x2=-1.2604

Export result? (Yes -> 1 \ No -> 0)
0

>solver(1/4x+5/3x-12)
#0=6.26087

>solve equation(1/4x+5/3x-12)
x1=6.26087

Export result? (Yes -> 1 \ No -> 0)
0

>solve equation(2x^2-7x-12+3x^3)
x1=1.82098
x2=-1.24382-0.80593i
x3=-1.24382+0.80593i

Export result? (Yes -> 1 \ No -> 0)
0

>solve equation(2x^2-7x-12+3x^3+2x^3)
x1=1.53351
x2=-0.966753-0.793997i
x3=-0.966753+0.793997i

Export result? (Yes -> 1 \ No -> 0)
0

>solve equation(5\2\_7\_12)

x1=1.53351
x2=-0.966753-0.793997i
x3=-0.966753+0.793997i

Export result? (Yes -> 1 \ No -> 0)
0

>
```

```
>solve  equation((((x-8)(x-9)(x-12))/(x-8)+((x-12)(x-20)(x-24))/(x-12))(x-2))

x1=2
x2=16.25-5.47152i
x3=16.25+5.47152i
```

Graph with table view



```
Current settings:

| Xmin: -10 | Xmax: 10 | Xscale: 1 | Ymin: -1 | Ymax: 1 | Yscale: 0.1 |

Legend:

sin(x) -> X
0.5cos(x) -> Y

Do you want to see the graph data in table form? (Yes -> 1 \ No -> 0)
1

|x           |X           |Y
-10          0.529919     -0.424024
-9.83333     0.390731     -0.460252
-9.66667     0.224951     -0.487185
-9.5         0.0697565    -0.498782
-9.33333     -0.104528    -0.497261
-9.16667     -0.258819    -0.482963
-9           -0.422618    -0.453154
-8.83333     -0.559193    -0.414519
-8.66667     -0.694658    -0.35967
-8.5         -0.798636    -0.300908
-8.33333     -0.891007    -0.226995
-8.16667     -0.956305    -0.146186
-8           -0.990268    -0.0695866
-7.83333     -0.999391    0.0174497
-7.66667     -0.981627    0.0954045
-7.5         -0.93358     0.179184
-7.33333     -0.866025    0.25
-7.16667     -0.766044    0.321394
-7           -0.656059    0.377355
-6.83333     -0.515038    0.428584
-6.66667     -0.358368    0.46679
-6.5         -0.207912    0.489074
-6.33333     -0.0348995   0.499695
-6.16667     0.121869     0.496273
-6           0.292372     0.478152
-5.83333     0.438371     0.449397
-5.66667     0.587785     0.404508
-5.5         0.707107     0.353553
-5.33333     0.819152     0.286788
-5.16667     0.898794     0.219186
-5           0.961262     0.137819
-4.83333     0.994522     0.0522642
-4.66667     0.99863      -0.026168
-4.5         0.97437      -0.112476
-4.33333     0.927184     -0.187303
-4.16667     0.848048     -0.26496
-4           0.75471      -0.32803
-3.83333     0.62932      -0.388573
-3.66667     0.5          -0.433013
-3.5         0.34202      -0.469846
-3.33333     0.173648     -0.492404
-3.16667     0.0174524    -0.499924
-3           -0.156434    -0.493844
-2.83333     -0.309017    -0.475528
-2.66667     -0.469472    -0.441474
-2.5         -0.601815    -0.399318
-2.33333     -0.731354    -0.340999
-2.16667     -0.829038    -0.279596
-2           -0.913545    -0.203368
-1.83333     -0.965926    -0.12941
-1.66667     -0.996195    -0.0435779
-1.5         -0.996195    0.0435779
-1.33333     -0.970296    0.120961
-1.16667     -0.913545    0.203368
-1           -0.838671    0.27232
-0.833333    -0.731354    0.340999
-0.666667    -0.615661    0.394005
-0.5         -0.469472    0.441474
-0.333333    -0.325568    0.472759
-0.166667    -0.156434    0.493844
-1.4988E-15  0            0.5
0.166667     0.156434     0.493844
0.333333     0.325568     0.472759
0.5          0.469472     0.441474
0.666667     0.615661     0.394005
0.833333     0.731354     0.340999
1            0.838671     0.27232
1.16667      0.913545     0.203368
```



Triangles Rectangles Solver

```
>triangles rectangles solver

This solver will give to you all trigonometrics unknowns in any triangle rectangle.

If you don't know how to answer to a request, please enter '_1' and hit 'Enter';
If you know how to answer, enter the answer and hit 'Enter';

You must enter angles in degrees because, if not, you will get wrong results.

Let's start!

Enter the hypotenuse value: 1000
Enter the angle formed by the hypotenuse with the adjacent: 30


Report of results

The percentage of slope is equal to 57.735027 percent;
The adjacent is equal to 866.025;
The opposite is equal to 500;
The hypotenuse is equal to 1000;
The angle formed by the hypotenuse with the adjacent is equal to 30;
The angle formed by the hypotenuse with the opposite is equal to 60.



Do you want to export the report? (Yes -> 1 / No -> 0)
1

Drag-and-drop to here your txt file to save the report.
"C:\Users\Renato Freitas\Desktop\relatorio.txt"

==> Your report was saved successfully. <==

Do you want to analyze more some triangles rectangles?
(Yes -> 1 / No -> 0)
0

>
```

Arithmetic Matrix Solver

```
What to do?
Sum of Matrices -> 1
Subtraction of Matrices -> 2
Multiply a Matrix by a complex/real number -> 3
Multiplication of Matrices -> 4
Transpose a Matrix -> 5
Inverse a Matrix -> 6
Power a Matrix -> 7
Matrix Rank -> 8
8


Matrix:
4\6\12;6\12\9;5\12\20

Matrix Rank:
4+0i 6+0i 12+0i
0+0i 3+0i -9+0i
0+0i 0+0i 18.5+0i

Rank: 3

Export result? (Yes -> 1 \ No -> 0)
0

Continue? (Yes -> 1 / No -> 0)
0

>
```

Calculation of Determinant

```
>det(2\3\4;4\5\6;2\7\4)
#0=16

>det(2\3\4;4\5\6;2\7\4)+det(2\3\4;4\5\6;2\7\4)
#1=32

>det(2\3\4;4\5\6;2\7\4)*det(2\3\4;4\5\6;2\7\4)
#2=256

>det(2\3\4;4\5\6;2\7\4)/det(2\3\4;4\5\6;2\7\4)
#3=1

>
```

```
>det(2+7i\3\4;4\5-20i\6;2+5i\7\4-20i)
#0=76-2924i

>
```

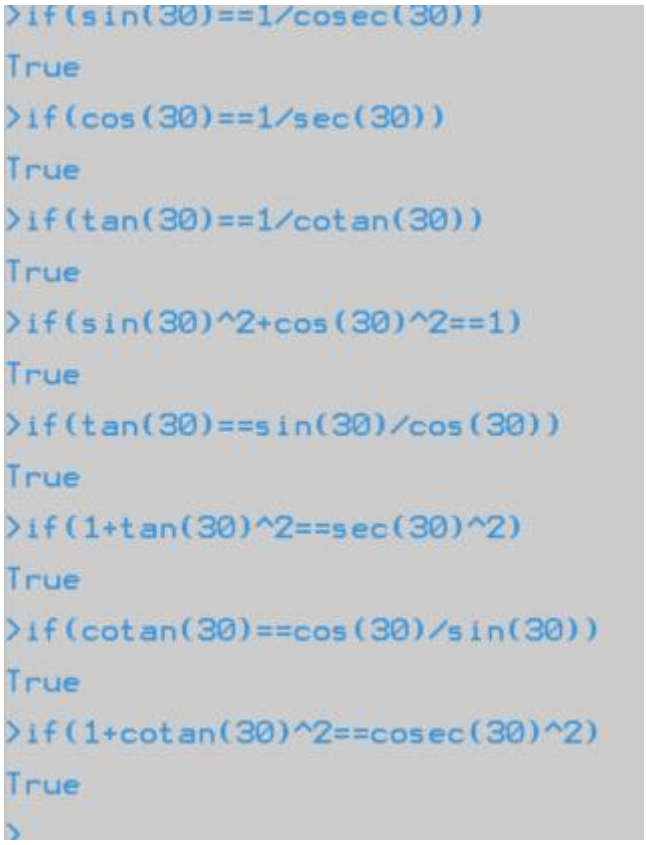
Roots to Polynomial

```
>roots to polynomial(4\12\3\4\6)

(1+0i)x^2+(-16+0i)x+(48+0i)
(1+0i)x^3+(-19+0i)x^2+(96+0i)x+(-144+0i)
(1+0i)x^4+(-23+0i)x^3+(172+0i)x^2+(-528+0i)x+(576+0i)

Final polynomial:
(1+0i)x^5+(-29+0i)x^4+(310+0i)x^3+(-1560+0i)x^2+(3744+0i)x+(-3456+0i)

Export result? (Yes -> 1 \ No -> 0)
0

>solve equation(1\_29\310\_1560\3744\_3456)

x1=12
x2=4+1.23076E-07i
x3=4+4.10999E-07i
x4=6
x5=3

Export result? (Yes -> 1 \ No -> 0)
0

>
```

If command

Returns True or False.

For example, check below trigonometry equalities:



Example of expressions

```
>if(sin(30)!=1/2)
False
>if(sin(60)!=sqrt(3)/2)
False
>if(true&&false)
False
>if(2==65&&4/2==2)
False
>if(4/2==2)
True
>if(true||true)
True
>if(true||false)
True
>if(false||true)
True
>if(false||false)
False
>if(true&&true)
True
>if(false&&false)
False
>if(false&&true)
False
>if(false&&true)
False
```

```
>if(sin(30)^2+cos(30)^2==1==true)
True
>if(true==sin(30)^2+cos(30)^2==1)
True
>if(false==sin(30)^2+cos(30)^2==1)
False
>if(false==sin(30)^2+cos(30)^2==2)
True
>if(sin(30)^2+cos(30)^2==1)
True
>if(!(((sin(30)!=0.5)!=(sin(30)!=0.5))==false))
False
>if((((sin(30)!=0.5)!=(sin(30)!=0.5))==false))
True
>if((((sin(30)!=0.5)!=(sin(30)==0.5))==false))
False
>if((((sin(30)==0.5)!=(sin(30)==0.5))==false))
True
>if((((sin(30)==0.5)==(sin(30)==0.5))==false))
False
>if((((sin(30)==0.5)==(sin(30)==0.5))==true))
True
>
```

Available operators:

== - is equal to

!= - is not equal to

<= - is smaller than or equal to

>= - is greater than or equal to

> - is greater than

< - is smaller than

&& - and

|| - or

! - not

Available logic words: “true” and “false”

Financial Calculations

```
>financial calculations

What to calculate?

Annual Percentage Yield -> 5                    Annuity (FV)- Solve for n -> 25          Annuity (PV)- Solve for n -> 19
Annuity Due Payment - FV -> 42                  Annuity Due Payment - PV -> 43           Annuity Payment (FV) -> 41
Annuity Payment (PV) -> 20                      Annuity Payment Factor - PV -> 37        Asset to Sales Ratio -> 45
Asset Turnover Ratio -> 46                      Average Collection Period -> 47

Balloon Balance of a Loan -> 35                 Bid Ask Spread -> 48                     Bond Equivalent Yield -> 49
Book Value per Share -> 50

Capital Asset Pricing Model -> 51               Capital Gains Yield -> 52                Compound Interest -> 4
Continuous Compounding -> 6                     Contribution Margin -> 53                Current Ratio -> 54
Current Yield -> 55

Days in Inventory -> 56                         Debt Coverage Ratio -> 57                Debt Ratio -> 58
Debt to Equity Ratio (D/E) -> 59                Debt to Income Ratio -> 60               Diluted Earnings per Share -> 61
Discounted Payback Period -> 18                 Dividend Payout Ratio -> 62              Dividend Yield (Stock) -> 63
Dividends Per Share -> 64                       Doubling Time -> 27                      Doubling Time - Continuous Compounding -> 29
Doubling Time - Simple Interest -> 26

Earnings Per Share -> 65                        Equity Multiplier -> 66                  Equivalent Annual Annuity -> 17
Estimated Earnings -> 67

Free Cash Flow to Equity -> 68                  Free Cash Flow to Firm (FCFF) -> 69      Future Value -> 2
Future Value of Annuity -> 24                   Future Value of Annuity Due -> 44        Future Value of Growing Annuity -> 38
FV of Annuity - Continuous Compounding -> 32    FV - Continuous Compounding -> 15        Future Value Factor -> 14

Geometric Mean Return -> 7                      Growing Annuity Payment - FV -> 39       Growing Annuity Payment - PV -> 40

Holding Period Return -> 8

Interest Coverage Ratio -> 71                   Inventory Turnover Ratio -> 72

Loan Payment -> 34                              Loan to Deposit Ratio -> 73              Loan to Value Ratio -> 74

Net Asset Value -> 75                           Net Present Value -> 10                  Net Profit Margin -> 76
Net Working Capital -> 77

Payback Period -> 16                            Payments on a Balloon Loan -> 36         Present Value -> 1
Present Value Factor -> 9                       Present Value of Annuity -> 11           Present Value Annuity Factor -> 21
Present Value of Annuity Due -> 22              PV of a Growing Annuity -> 23            PV of Growing Perpetuity -> 70
PV - Continuous Compounding -> 12               PV of Perpetuity -> 78                   Preferred Stock -> 79
Price to Book Value -> 80                       Price to Earnings Ratio -> 81            Price to Sales Ratio -> 82

Quick Ratio -> 83

Rate of Inflation -> 84                         Real Rate of Return -> 85                Receivables Turnover Ratio -> 86
Retention Ratio -> 87                           Return on Assets -> 88                   Return on Equity (ROE) -> 89
Return on Investment -> 31                      Remaining Balance on Loan -> 33          Risk Premium -> 90
Rule of 72 -> 28

Solve for Number of Periods - PV & FV -> 13     Simple Interest -> 3                     Stock - PV with Constant Growth -> 91
Stock - PV with Zero Growth -> 92

Tax Equivalent Yield -> 93                      Total Stock Return -> 94

Weighted Average -> 30

Yield to Maturity -> 95

Zero Coupon Bond Value -> 96                    Zero Coupon Bond Effective Yield -> 97
```

Statistics Calculations

```
>statistics calculations

What to calculate?

Population Measures (Mean, Variance and Standard Deviation) -> 1
Sampling (Sample mean, Sample variance,  Standard Deviation) -> 2

Number Sequence

Arithmetic Sequence -> 3
Geometric Sequence -> 4
Fibonacci Sequence -> 5

Sample Size

Find out the sample size -> 6
Find out the confidence interval -> 7

Probability

Probability of Two Events -> 8
Probability of a Normal Distribution -> 9

Permutation and Combination -> 10


Mean, Median, Mode and Range -> 11

11

Population?
Info: separate values by using '\'. Example: "20\30\60"
3\4\5\12\5

Mean: 5.8
Median: 5
Mode: 5 -> appeared 2 times
Range: 9
Largest: 12
Smallest: 3
Sum: 29
Count 5
Continue? (Yes -> 1 / No -> 0)
0

>_
```

Valid Confidence Level values for the option: "Find out the confidence interval"	z-score (±)
0.70	1.04
0.75	1.15
0.80	1.28
0.85	1.44
0.92	1.75
0.95	1.96
0.96	2.05
0.98	2.33
0.99	2.58
0.999	3.29

0.9999	3.89
0.99999	4.42

Geometry Calculations

```
>geometry calculations

What to calculate?

Areas and Perimeters:

Square -> 1
Rectangle -> 2
Parallelogram -> 3
Trapezoid -> 4
Triangle -> 5
Circle -> 6

Volumes and Surface Areas of Three-Dimensional Figures:

Rectangular Solid -> 7
Cube -> 8
Right Circular Cylinder -> 9
Sphere -> 10
Right Circular Cone -> 11
Square or Rectangular Pyramid -> 12
Frustum of Right Circular Cone -> 13
10

Radius?
4

Volume: 268.083
Surface Area: 201.062

Continue? (Yes -> 1 / No -> 0)
0

>
```

Physics Calculations

```
>physics calculations

What to calculate?

Acceleration -> 1                                   Buoyancy Force -> 2
Centripetal Force -> 3                              Elastic Collision -> 4
Density -> 5                                        Displacement -> 6
Free fall distance without air resistance -> 7      Free fall velocity without air resistance -> 8
Free fall distance with air resistance -> 9         Free fall velocity with air resistance -> 10
Newton's 2nd law -> 11                              Friction -> 12
Weight/Force Mass Gravity -> 13                     Gravitational Force -> 14
Impact Force -> 15                                  Kinematics -> 16
Lever Force -> 17                                   Trajectory -> 18
Momentum -> 19                                      Simple Pendulum -> 20
Projectile Motion -> 21                             Resultant Vector -> 22
Torque -> 23                                        Velocity -> 24
Average Velocity -> 25
25
Enter the unknown value as 'x'
Initial Velocity (m/s)?
12
Velocity (m/s)?
79
Average Velocity (m/s)?
x

Average Velocity: 45.5 (m/s)

Continue? (Yes -> 1 / No -> 0)
0

>
```

Unit Conversions

```
>unit conversions

What to convert?

Length -> 1
Temperature -> 2
Area -> 3
Volume -> 4
Weight -> 5
Time -> 6
Angle -> 7
3


Your unit?

Square Meter -> 1
Square Kilometer -> 2
Square Centimeter -> 3
Square Millimeter -> 4
Square Micrometer -> 5
Hectare -> 6
Square Mile -> 7
Square Yard -> 8
Square Foot -> 9
Square Inch -> 10
Acre -> 11
6

Value?
1

Square Meter: 10000
Square Kilometer: 0.01
Square Centimeter: 1E+08
Square Millimeter: 1E+10
Square Micrometer: 1E+16
Hectare: 1
Square Mile: 0.00386102
Square Yard: 11959.9
Square Foot: 107639
Square Inch: 1.55E+07
Acre: 2.47105

Continue with Area conversions? (Yes -> 1 / No -> 0)
0

Continue? (Yes -> 1 / No -> 0)
0

>_
```

Microeconomics Calculations

```
>microeconomics calculations

What to calculate?

Midpoint Method for Price Elasticity of Demand -> 1
Average Fixed Cost -> 2
Average Variable cost -> 3
Average Total cost -> 4
Unit Cost / Average Total cost -> 5
Profit -> 6
Profit based on Price, Avg Unit Cost and Quantity -> 7
Economic Profit -> 8
Cross-Price Elasticity of Demand -> 9
Income Elasticity of Demand -> 10
Price Elasticity of Demand -> 11
Consumer Surplus -> 12
Producer Surplus -> 13
Price Elasticity of Supply -> 14
Total Surplus -> 15
Accounting Profit -> 16
6

Total Revenue?
123
Total Expenses?
100

Profit: 23

Continue? (Yes -> 1 / No -> 0)
0

>_
```

Polynomial Simplifier

Division

```
>simplify polynomial((((x-7)(x-8)(x-12))/((x-7)(x-8)))/(((x-7)(x-8)(x-12))/((x-7)(x-8))))
((0+0i)x+(1+0i))
```

Multiplication

```
>simplify polynomial((((x-7)(x-8)(x-12))/((x-7)(x-8)))((((x-7)(x-8)(x-12))/((x-7)(x-8))))
((1+0i)x^2+(-24+0i)x+(144+0i))
```

Addition

```
>simplify polynomial((((x-7)(x-8)(x-12))/((x-7)(x-8)))+(((x-7)(x-8)(x-12))/((x-7)(x-8))))
((2+0i)x+(-24+0i))
```

Subtraction

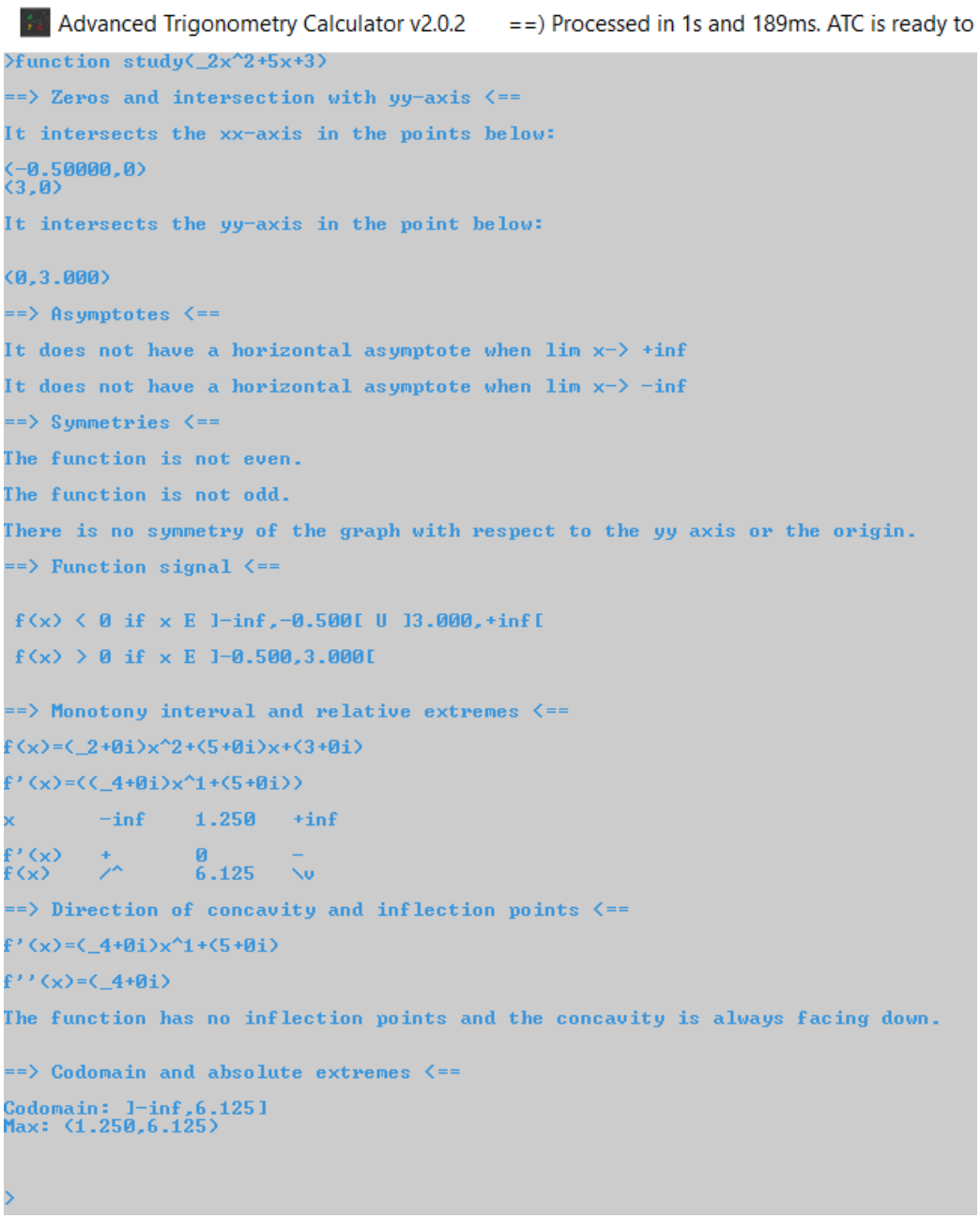
```
>simplify polynomial((((x-7)(x-8)(x-12))/((x-7)(x-8)))-(((x-7)(x-8)(x-12))/((x-7)(x-8))))

((0+0i)x+(0+0i))
```

Complex simplification

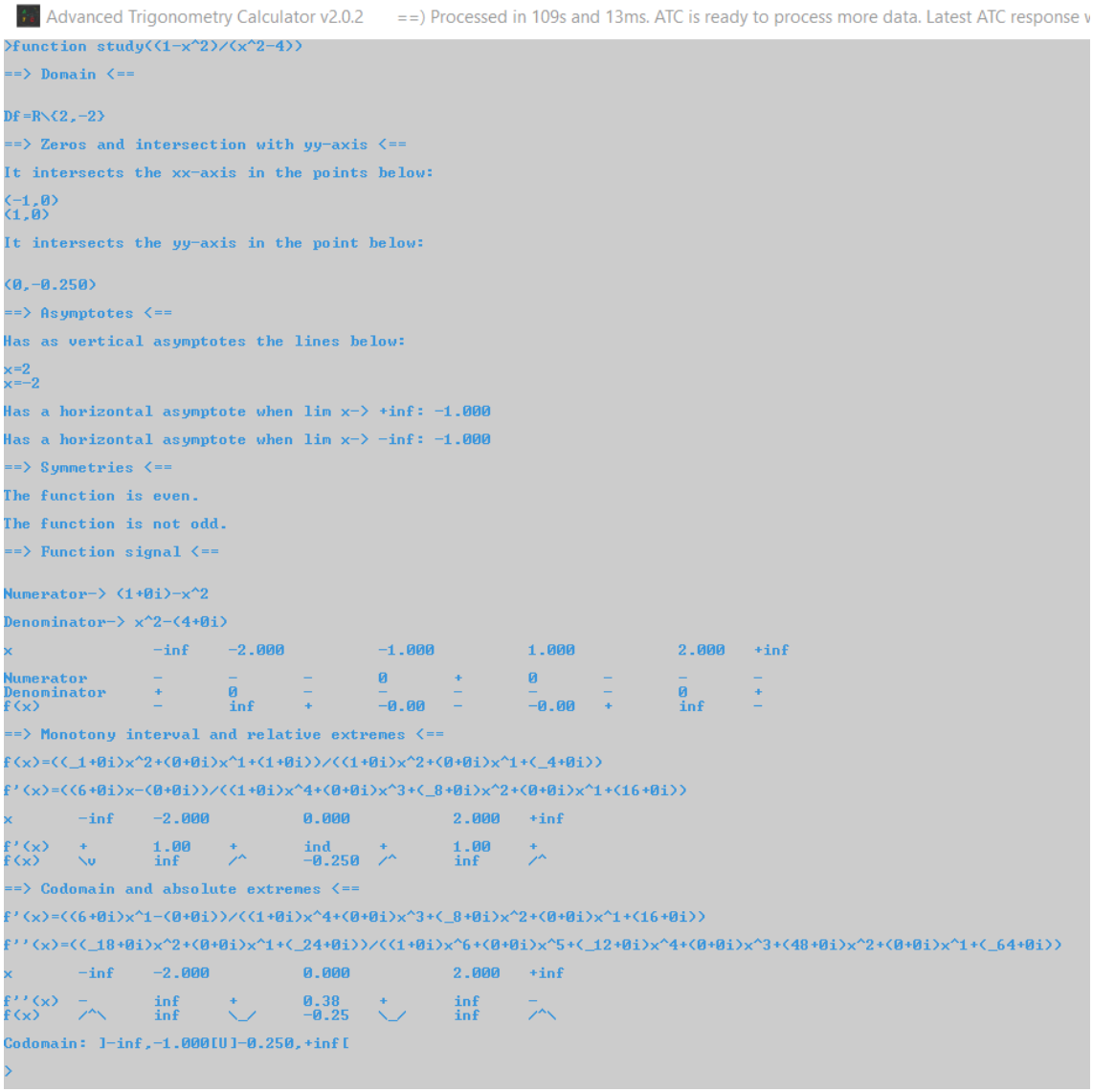
```
>simplify polynomial(((((x-7)(x-8)(x-12))/((x-7)(x-8)))(((x-7)(x-8)(x-12))/((x-7)(x-8))))/((((x-7)(x
-8)(x-12))/((x-7)(x-8)))(((x-7)(x-8)(x-12))/((x-7)(x-8)))))

((0+0i)x+(1+0i))
```

Function Study

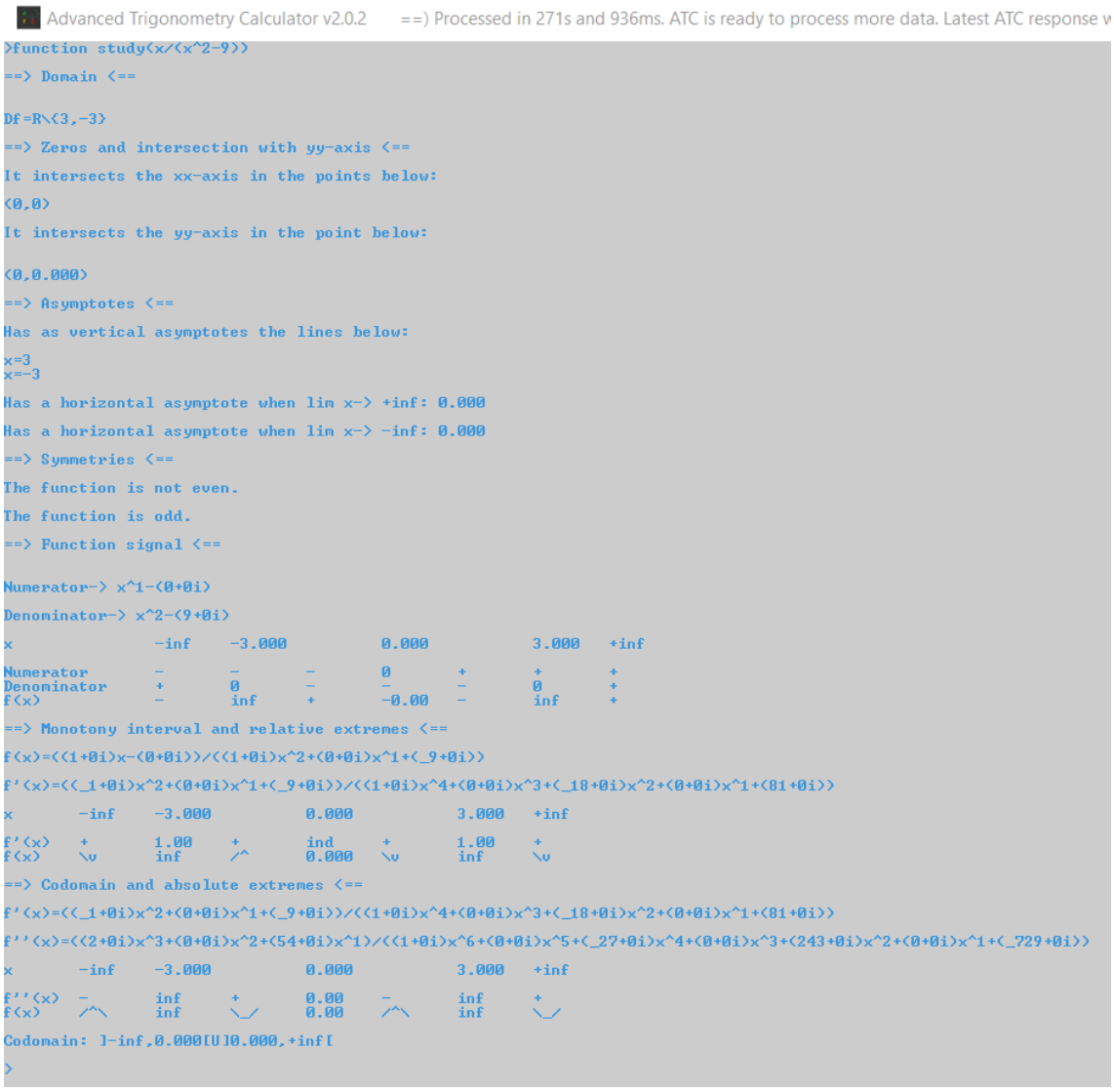
Example 1



Example 2


Advanced Trigonometry Calculator v2.0.2 ==) Processed in 109s and 13ms. ATC is ready to process more data. Latest ATC response v
>function study<(1-x^2)/(x^2-4))
==> Domain <==
Df=R\{2,-2}
==> Zeros and intersection with yy-axis <==
It intersects the xx-axis in the points below:
(-1,0)
(1,0)
It intersects the yy-axis in the point below:
(0,-0.250)
==> Asymptotes <==
Has as vertical asymptotes the lines below:
x=2
x=-2
Has a horizontal asymptote when lim x-> +inf: -1.000
Has a horizontal asymptote when lim x-> -inf: -1.000
==> Symmetries <==
The function is even.
The function is not odd.
==> Function signal <==
Numerator-> (1+0i)-x^2
Denominator-> x^2-(4+0i)
x -inf -2.000 -1.000 1.000 2.000 +inf
Numerator - - - 0 + 0 - - -
Denominator + 0 - - - - 0 +
f(x) - inf + -0.00 - -0.00 + inf -
==> Monotony interval and relative extremes <==
f(x)=((_1+0i)x^2+(0+0i)x^1+(1+0i))/((1+0i)x^2+(0+0i)x^1+(_4+0i))
f'(x)=((6+0i)x-(0+0i))/((1+0i)x^4+(0+0i)x^3+(_8+0i)x^2+(0+0i)x^1+(16+0i))
x -inf -2.000 0.000 2.000 +inf
f'(x) + 1.00 + ind + 1.00 +
f(x) \v inf /^ -0.250 /^ inf /^
==> Codomain and absolute extremes <==
f'(x)=((6+0i)x^1-(0+0i))/((1+0i)x^4+(0+0i)x^3+(_8+0i)x^2+(0+0i)x^1+(16+0i))
f''(x)=((_18+0i)x^2+(0+0i)x^1+(_24+0i))/((1+0i)x^6+(0+0i)x^5+(_12+0i)x^4+(0+0i)x^3+(48+0i)x^2+(0+0i)x^1+(_64+0i))
x -inf -2.000 0.000 2.000 +inf
f''(x) - inf + 0.38 + inf -
f(x) /^\ inf _/ -0.25 _/ inf /^\
Codomain:]-inf,-1.000[U]-0.250,+inf[
>

Example 3


Advanced Trigonometry Calculator v2.0.2 ==) Processed in 271s and 936ms. ATC is ready to process more data. Latest ATC response w
>function study(x/(x^2-9))
==> Domain <==
Df=R\{3,-3}
==> Zeros and intersection with yy-axis <==
It intersects the xx-axis in the points below:
(0,0)
It intersects the yy-axis in the point below:
(0,0.000)
==> Asymptotes <==
Has as vertical asymptotes the lines below:
x=3
x=-3
Has a horizontal asymptote when lim x-> +inf: 0.000
Has a horizontal asymptote when lim x-> -inf: 0.000
==> Symmetries <==
The function is not even.
The function is odd.
==> Function signal <==
Numerator-> x^1-(0+0i)
Denominator-> x^2-(9+0i)
x -inf -3.000 0.000 3.000 +inf
Numerator - - - 0 + + +
Denominator + 0 - - - 0 +
f(x) - inf + -0.00 - inf +
==> Monotony interval and relative extremes <==
f(x)=((1+0i)x-(0+0i))/((1+0i)x^2+(0+0i)x^1+(_9+0i))
f'(x)=((_1+0i)x^2+(0+0i)x^1+(_9+0i))/((1+0i)x^4+(0+0i)x^3+(_18+0i)x^2+(0+0i)x^1+(81+0i))
x -inf -3.000 0.000 3.000 +inf
f'(x) + 1.00 + ind + 1.00 +
f(x) \u inf /^ 0.000 \u inf \u
==> Codomain and absolute extremes <==
f'(x)=((_1+0i)x^2+(0+0i)x^1+(_9+0i))/((1+0i)x^4+(0+0i)x^3+(_18+0i)x^2+(0+0i)x^1+(81+0i))
f''(x)=((2+0i)x^3+(0+0i)x^2+(54+0i)x^1)/((1+0i)x^6+(0+0i)x^5+(_27+0i)x^4+(0+0i)x^3+(243+0i)x^2+(0+0i)x^1+(_729+0i))
x -inf -3.000 0.000 3.000 +inf
f''(x) - inf + 0.00 - inf +
f(x) /^\ inf _/ 0.00 /^\ inf _/
Codomain:]-inf,0.000[U]0.000,+inf[
>

Reduction to the same denominator

```
>solver((x-7)/(x+6)+(x-12)/(x+9)-(x-24)/(x+5))
#4=-6.48337

>x=#4
#5=-6.48337

>((x-7)/(x+6)+(x-12)/(x+9)-(x-24)/(x+5))
#6=-1.37469E-05

>simplify polynomial((x-7)/(x+6)+(x-12)/(x+9)-(x-24)/(x+5))

((1+0i)x^3+(15.000001+0i)x^2+(150.999992+0i)x^1+(620.999931+0i))


Export result? (Yes -> 1 \ No -> 0)
0

>solver((1+0i)x^3+(15.000001+0i)x^2+(150.999992+0i)x^1+(620.999931+0i))
#7=-6.48337
```

FFT and IFFT

```
>fft(1+2i\2+3i\3+4i\4+5i\5+6i)

X[0]=15+20i
X[1]=-5.940955+0.940955i
X[2]=-3.312299-1.687701i
X[3]=-1.687701-3.312299i
X[4]=0.940955-5.940955i


>ifft(15+20i\_5.940955+0.940955i\_3.312299-1.687701i\_1.687701-3.312299i\0.940955-5.940955i)

x[0]=1+2i
x[1]=2+3i
x[2]=3+4i
x[3]=4+5i
x[4]=5+6i


>
```

ATC as Arithmetic Matrix Calculator

```
>matrix=7\8\9;6\7\9;2\4\8

>matrix
#0=
7+0i  8+0i  9+0i
6+0i  7+0i  9+0i
2+0i  4+0i  8+0i

>matrix^T
#1=
7+0i  6+0i  2+0i
8+0i  7+0i  4+0i
9+0i  9+0i  8+0i

>matrix^2
#2=
115+0i  148+0i  207+0i
102+0i  133+0i  189+0i
54+0i  76+0i  118+0i

>matrix*matrix
#3=
115+0i  148+0i  207+0i
102+0i  133+0i  189+0i
54+0i  76+0i  118+0i

>matrix-matrix
#4=
0+0i  0+0i  0+0i
0+0i  0+0i  0+0i
0+0i  0+0i  0+0i

>matrix*2
#5=
14+0i  16+0i  18+0i
12+0i  14+0i  18+0i
4+0i  8+0i  16+0i
```

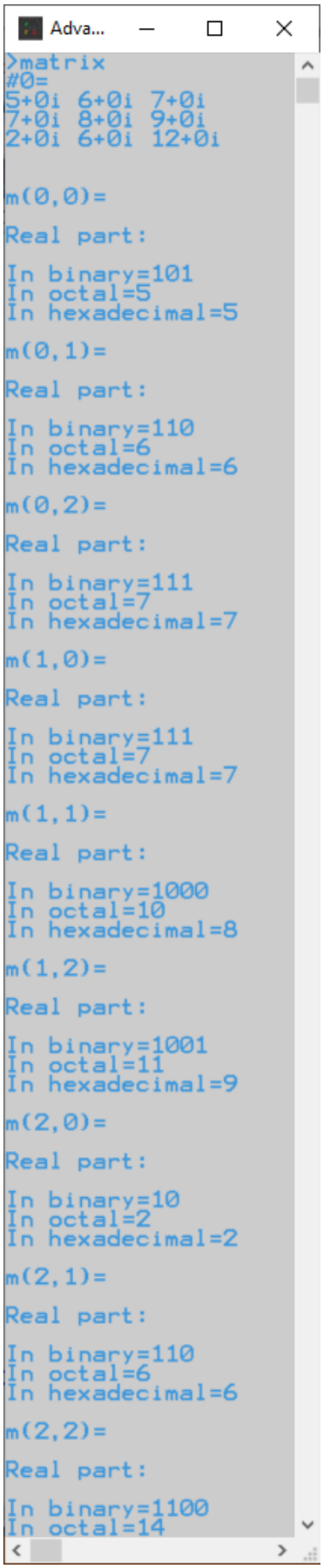
```
>matrix=matrix/2
3.5+0i  4+0i  4.5+0i
3+0i  3.5+0i  4.5+0i
1+0i  2+0i  4+0i

Export result? (Yes -> 1 \ No -> 0)
0

Export result? (Yes -> 1 \ No -> 0)
0
#7=0

>matrix
3.5+0i  4+0i  4.5+0i
3+0i  3.5+0i  4.5+0i
1+0i  2+0i  4+0i

Export result? (Yes -> 1 \ No -> 0)
0
#8=0

>matrix=matrix*2
7+0i  8+0i  9+0i
6+0i  7+0i  9+0i
2+0i  4+0i  8+0i

Export result? (Yes -> 1 \ No -> 0)
0

Export result? (Yes -> 1 \ No -> 0)
0
#9=0

>matrix
7+0i  8+0i  9+0i
6+0i  7+0i  9+0i
2+0i  4+0i  8+0i

Export result? (Yes -> 1 \ No -> 0)
0
#10=0

>
```

```
>newmatrix=7\8\9;4\5\6;8\9\7
#1=80907

>matrix
7+0i  8+0i  9+0i
6+0i  7+0i  9+0i
2+0i  4+0i  8+0i

Export result? (Yes -> 1 \ No -> 0)
0
#2=0

>det(newmatrix)
7+0i  8+0i  9+0i
4+0i  5+0i  6+0i
8+0i  9+0i  7+0i

Export result? (Yes -> 1 \ No -> 0)
0
#3=-9

>det(matrix)
7+0i  8+0i  9+0i
6+0i  7+0i  9+0i
2+0i  4+0i  8+0i

Export result? (Yes -> 1 \ No -> 0)
0
#4=-10

>matrix^R
7+0i  8+0i  9+0i
6+0i  7+0i  9+0i
2+0i  4+0i  8+0i

Export result? (Yes -> 1 \ No -> 0)
0
#5=3

>newmatrix^R
7+0i  8+0i  9+0i
4+0i  5+0i  6+0i
8+0i  9+0i  7+0i

Export result? (Yes -> 1 \ No -> 0)
0
#6=3
```

```
>matrix^_3
-82.97+0i 108.348+0i -27.999+0i
114.57+0i -149.588+0i 38.619+0i
-35.83+0i 46.772+0i -12.061+0i

Export result? (Yes -> 1 \ No -> 0)
0
#7=0

>matrix^_1*matrix^_1*matrix^_1
-82.97+0i 108.348+0i -27.999+0i
114.57+0i -149.588+0i 38.619+0i
-35.83+0i 46.772+0i -12.061+0i

Export result? (Yes -> 1 \ No -> 0)
0
#8=0

>matrix^_1
-2+0i 2.8+0i -0.9+0i
3+0i -3.8+0i 0.9+0i
-1+0i 1.2+0i -0.1+0i

Export result? (Yes -> 1 \ No -> 0)
0
#9=0

>newsecondmatrix=matrix^_1
-2+0i 2.8+0i -0.9+0i
3+0i -3.8+0i 0.9+0i
-1+0i 1.2+0i -0.1+0i

Export result? (Yes -> 1 \ No -> 0)
0

Export result? (Yes -> 1 \ No -> 0)
0
#10=0

>newsecondmatrix^_1
7+0i 8+0i 9+0i
6+0i 7+0i 9+0i
2+0i 4+0i 8+0i

Export result? (Yes -> 1 \ No -> 0)
0
#11=0

>
```

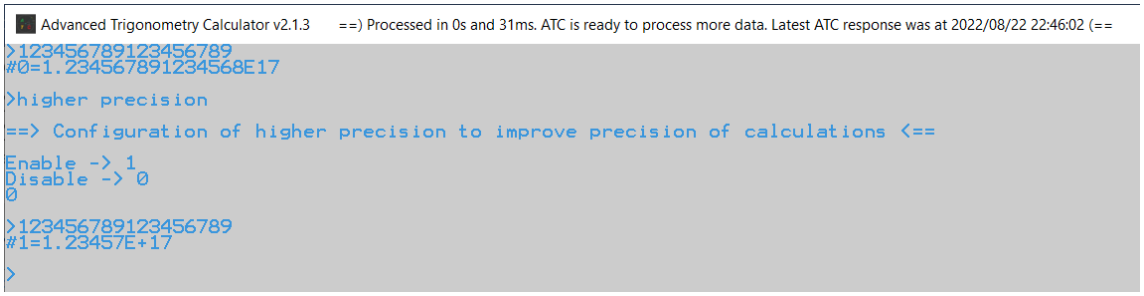
```
>matrixone=5\6;9\12

>(matrixone^_1)^_1
#7=
5.00002+0i 6.00002+0i
9.00004+0i 12+0i

>matrix^_2
#8=
13.3+0i -17.32+0i 4.41+0i
-18.3+0i 23.92+0i -6.21+0i
5.7+0i -7.48+0i 1.99+0i

>matrix^_1*matrix^_1
#9=
13.3+0i -17.32+0i 4.41+0i
-18.3+0i 23.92+0i -6.21+0i
5.7+0i -7.48+0i 1.99+0i

>(2*matrix)/matrix
#10=2

>matrix
#11=
7+0i 8+0i 9+0i
6+0i 7+0i 9+0i
2+0i 4+0i 8+0i
```

```
>sin(30)
#0=0.5

>matrix
#1=
7+0i 8+0i 9+0i
9+0i 5+0i 4+0i
2+0i 3+0i 4+0i


>matriz
#2=
6+0i 7+0i 8+0i 9+0i 12+0i 4+0i 7+0i
7+0i 8+0i 9+0i 0+0i 12+0i 1+0i 3+0i
3+0i 6+0i 7+0i 8+0i 2+0i 9+0i 12+0i


>tan(30)
#3=0.57735

>see results

Result value

#0=0.5
#1=
7+0i 8+0i 9+0i
9+0i 5+0i 4+0i
2+0i 3+0i 4+0i

#2=
6+0i 7+0i 8+0i 9+0i 12+0i 4+0i 7+0i
7+0i 8+0i 9+0i 0+0i 12+0i 1+0i 3+0i
3+0i 6+0i 7+0i 8+0i 2+0i 9+0i 12+0i

#3=0.57735

>
```


Adva...
>matrix
#0=
5+0i 6+0i 7+0i
7+0i 8+0i 9+0i
2+0i 6+0i 12+0i
m(0,0)=
Real part:
In binary=101
In octal=5
In hexadecimal=5
m(0,1)=
Real part:
In binary=110
In octal=6
In hexadecimal=6
m(0,2)=
Real part:
In binary=111
In octal=7
In hexadecimal=7
m(1,0)=
Real part:
In binary=111
In octal=7
In hexadecimal=7
m(1,1)=
Real part:
In binary=1000
In octal=10
In hexadecimal=8
m(1,2)=
Real part:
In binary=1001
In octal=11
In hexadecimal=9
m(2,0)=
Real part:
In binary=10
In octal=2
In hexadecimal=2
m(2,1)=
Real part:
In binary=110
In octal=6
In hexadecimal=6
m(2,2)=
Real part:
In binary=1100
In octal=14

```
>current settings

Mode-----------------------------------> Radian DEGREE Gradian | Info: Enter "mode" to change.
Numerical Systems Response-------------> Enabled DISABLED | Info: Enter "numerical systems" to change.
SI Prefixes Response-------------------> Enabled DISABLED | Info: Enter "si prefixes" to change.
Actual Time Response-------------------> Enabled DISABLED | Info: Enter "actual time response" to change.
Colors---------------------------------> Text: AQUA - Background: WHITE | Info: Enter "colors" to change.
Window---------------------------------> x: 15 - y: 0 - Width: 1361 - Height: 750 | Info: Enter "window" to change.
Dimensions-----------------------------> Lines: 5000 - Columns: 154 | Info: Enter "dimensions" to change.

>radsin(matrix)
#1=
0.831322+-0.587344i 0.990268+0i 0.422618+0i -0.544639+-0i
0.913545+0i 0.156434+0i -0.75471+-0i -0.961262+-0i


>matrix=0\15\30\45;60\75\90\105;120\135\150\165;180\195\210\225
#2=
0+0i 15+0i 30+0i 45+0i
60+0i 75+0i 90+0i 105+0i
120+0i 135+0i 150+0i 165+0i
180+0i 195+0i 210+0i 225+0i


>sin(matrix)
#3=
0+0i 0.258819+0i 0.5+0i 0.707107+0i
0.866025+0i 0.965926+0i 1+0i 0.965926+0i
0.866025+0i 0.707107+0i 0.5+0i 0.258819+0i
0+0i -0.258819+-0i -0.5+-0i -0.707107+-0i


>asin(ans)
#4=
0+1.27222E-14i 15+1.27222E-14i 30+1.27222E-14i 45+1.27222E-14i
60+1.27222E-14i 75+1.27222E-14i 90+1.27222E-14i 75+1.27222E-14i
60+1.27222E-14i 45+1.27222E-14i 30+1.27222E-14i 15+1.27222E-14i
0+1.27222E-14i -15+1.27222E-14i -30+1.27222E-14i -45+1.27222E-14i


>asin(#3)
#5=
0+1.27222E-14i 15+1.27222E-14i 30+1.27222E-14i 45+1.27222E-14i
60+1.27222E-14i 75+1.27222E-14i 90+1.27222E-14i 75+1.27222E-14i
60+1.27222E-14i 45+1.27222E-14i 30+1.27222E-14i 15+1.27222E-14i
0+1.27222E-14i -15+1.27222E-14i -30+1.27222E-14i -45+1.27222E-14i


>sin(#2)
#6=
0+0i 0.258819+0i 0.5+0i 0.707107+0i
0.866025+0i 0.965926+0i 1+0i 0.965926+0i
0.866025+0i 0.707107+0i 0.5+0i 0.258819+0i
0+0i -0.258819+-0i -0.5+-0i -0.707107+-0i


>
```

Higher precision to improve precision

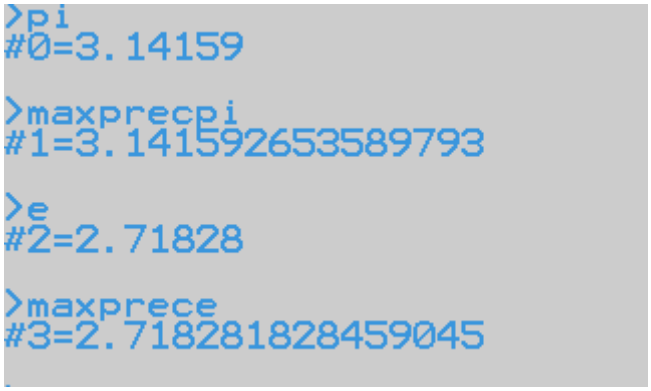


In this case this feature was enabled, and we disabled it.

Note that this feature when enabled is applied to almost all calculations performed with ATC.

Higher precision on demand by prefix

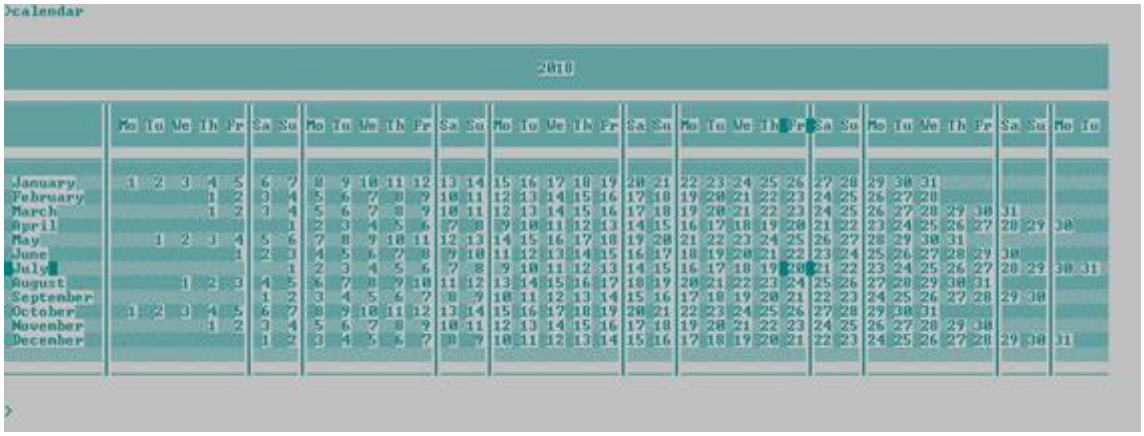
Using the prefix “maxprec” before an entire expression, you will be able get the result with up 15 decimal places without enable the higher precision feature mode. You will get the same result but as said on demand.



Time features

Calendar

Check calendars.



This example shows the calendar that corresponds to the “current year”. Note that you get the day, the month and the day of week highlighted.

Entering “calendar(2019)” you will get the calendar for 2019 but you will not get the day, the month and the day of week highlighted.

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.

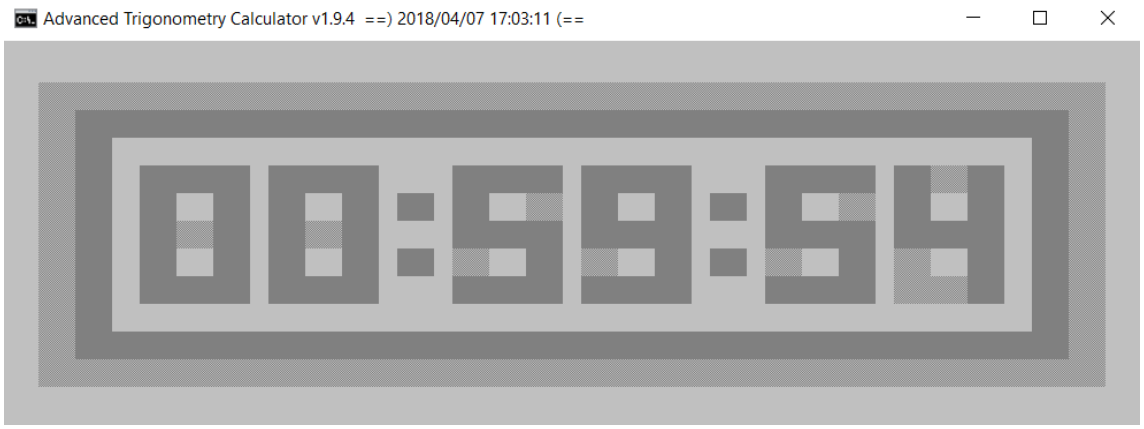
```
>timer(1:50:30)
```

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.

Big timer

Manage your time the big way.

```
>run big timer(::3600)
```



Date adjustment

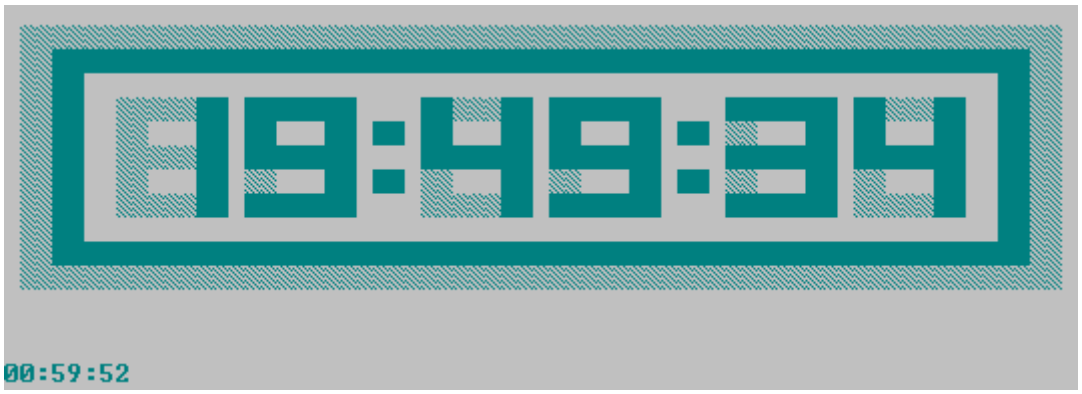
```
>day of week(d23+100 000m3+400y2015+500)
y2822m5d8=Sunday

>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)
```

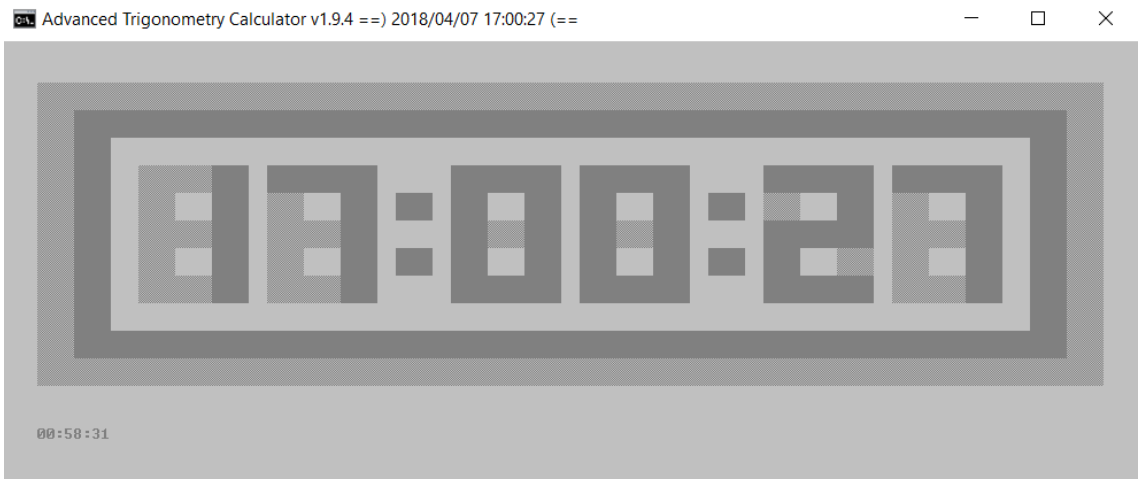


The example above shows a clock that will work during 1 hour.

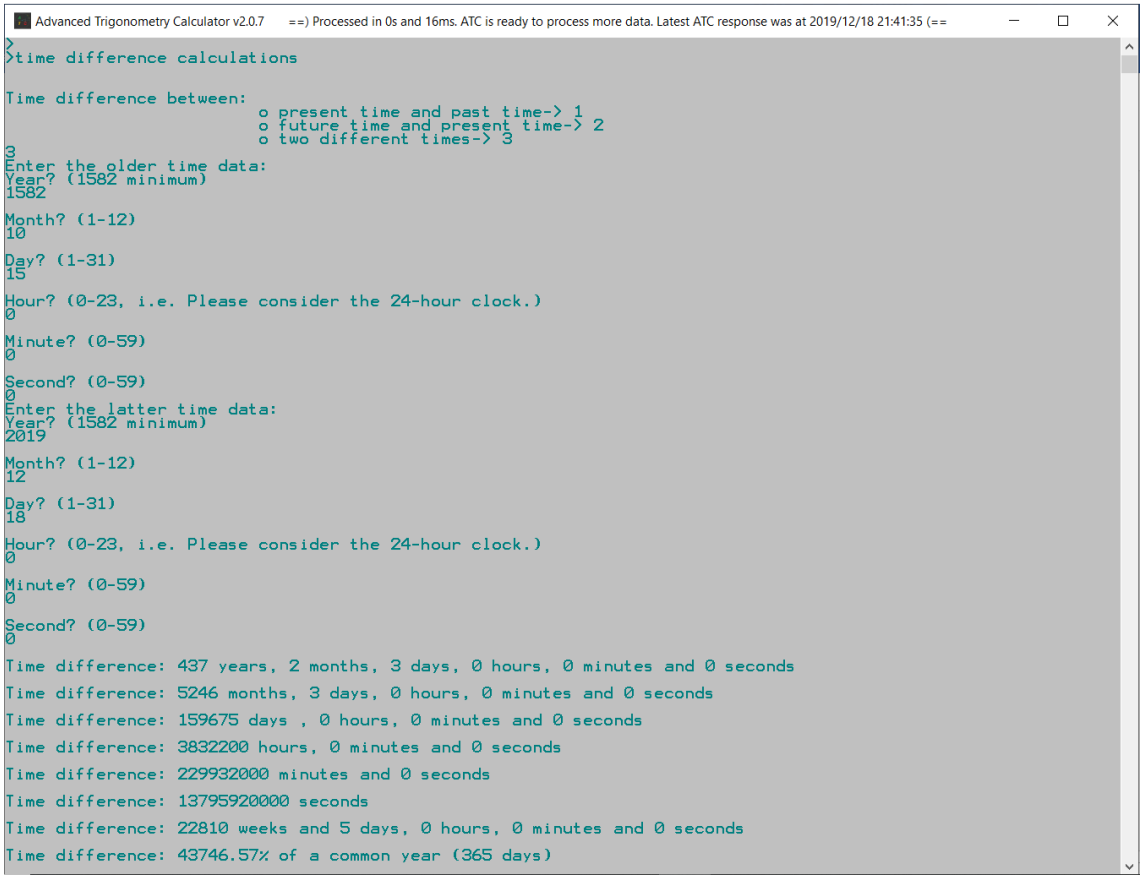
Big clock

Use a big clock.

```
run big clock(::3600)
```



Time Difference Calculations



Sorting

Ascending order

```
>ascending order(_4\_9\12\_1233\_67\123\_50000\1235465756\23\56\pi\e\_pi\e\_45356565676576)

-4.53566E+13, -50000, -1233, -67, -9, -4, -3.14159, 2.71828, 2.71828, 3.14159, 12, 23, 56, 123, 1.23547E+09


Export result? (Yes -> 1 \ No -> 0)
0
```

Descending order

```
>descending order(_4\_9\12\_1233\_67\123\_50000\1235465756\23\56\pi\e\_pi\e\_45356565676576)
1.23547E+09, 123, 56, 23, 12, 3.14159, 2.71828, 2.71828, -3.14159, -4, -9, -67, -1233, -50000, -4.53566E+13

Export result? (Yes -> 1 \ No -> 0)
0
```

ASCII order

```
>ascii order

Seperate your expressions by "\" and in the final press "Enter"

Austria\Italy\Belgium\Latvia\Bulgaria\Lithuania\Croatia\Luxembourg\Cyprus\Malta\Czechia\Netherlands\Denmark\Poland\Estonia\Portugal\Finland\Romania\France
\Slovakia\Germany\Slovenia\Greece\Spain\Hungary\Sweden\Ireland

Austria
Belgium
Bulgaria
Croatia
Cyprus
Czechia
Denmark
Estonia
Finland
France
Germany
Greece
Hungary
Ireland
Italy
Latvia
Lithuania
Luxembourg
Malta
Netherlands
Poland
Portugal
Romania
Slovakia
Slovenia
Spain
Sweden


Export result? (Yes -> 1 \ No -> 0)
0


Continue? (Yes -> 1 \ No -> 0)
0
```

Inverse ASCII order

```
>inverse ascii order

Seperate your expressions by "\" and in the final press "Enter"

Austria\Italy\Belgium\Latvia\Bulgaria\Lithuania\Croatia\Luxembourg\Cyprus\Malta\Czechia\Netherlands\Denmark\Poland\Estonia\Portugal\Finland\Romania\France
\Slovakia\Germany\Slovenia\Greece\Spain\Hungary\Sweden\Ireland

Sweden
Spain
Slovenia
Slovakia
Romania
Portugal
Poland
Netherlands
Malta
Luxembourg
Lithuania
Latvia
Italy
Ireland
Hungary
Greece
Germany
France
Finland
Estonia
Denmark
Czechia
Cyprus
Croatia
Bulgaria
Belgium
Austria


Export result? (Yes -> 1 \ No -> 0)
0


Continue? (Yes -> 1 \ No -> 0)
0
```

Download and contact

Setup Advanced Trigonometry Calculator x86.exe – installation file.

Setup Advanced Trigonometry Calculator x64.exe – installation file.

Advanced Trigonometry Calculator.zip – portable file.

Advanced Trigonometry Calculator Wine Linux.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.

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, swimming, programming, studying, socialize, watch documentaries of scientific and technological nature, discover new things, and he searches every time for more knowledge.