

MagicCalc 4.49 Product Manual

Publication Date: 08 February 2014

Copyright © HOUCINE ROMDHANE

Please check www.magiccalc.net periodically for product manual updates.

1 – INTERFACE DESCRIPTION:	3
2 – AVAILABLE WINDOWS:	4
2.1 – SWITCHING WINDOWS:.....	4
2.2 – CONSOLE WINDOW:.....	4
2.3 – PROGRAM WINDOW:.....	5
2.3.1 – <i>Presentation:</i>	5
2.4 – 2D WINDOW:	6
2.4.1 – <i>2D Window:</i>	6
2.4.2 – <i>2D Scale manipulation:</i>	7
2.5 – 3D WINDOW:	9
2.5.1 – <i>3D Window:</i>	9
2.5.2 – <i>3D Scale manipulation:</i>	9
3 – MAKING COMPUTATIONS	11
3.1 – MAKING COMPUTATIONS	11
3.2 – WORKING WITH VARIABLES.....	11
3.3 – USER INPUT.....	12
3.4 – USER OUTPUT.....	12
3.5 – BASE COMPUTATIONS	12
3.5.1 – <i>Bases computation keyboard:</i>	13
3.5.2 – <i>Bases logical operators:</i>	13
3.5.3 – <i>Bases conversion functions</i>	13
3.5.3 – <i>Base mode switching functions</i>	14
3.6 – SCIENTIFIC FUNCTIONS	14
3.6.1 – <i>Regular functions</i>	14
3.6.2 – <i>Operators:</i>	15
3.6.3 – <i>Constants:</i>	15
3.6.4 – <i>Utility functions:</i>	15
3.6.5 – <i>Trigonometric mode switching functions</i>	15
3.6.6 – <i>Trigonometric functions:</i>	16
3.6.7 – <i>Inverse trigonometric functions:</i>	16
3.6.8 – <i>Hyperbolic trigonometric functions:</i>	16
3.6.9 – <i>Inverse hyperbolic trigonometric functions:</i>	16
4 – MAKING GRAPHICS	17
4.1 – PRESENTATION	17
4.2 – 2D FUNCTIONS	17
4.3 – 2D PARAMETRIC FUNCTIONS:.....	18
4.4 – 3D FUNCTIONS:.....	19
4.5 – 3D PARAMETRIC FUNCTIONS:.....	20
4.6 – GRAPHING LIMITATIONS:	21
5 - PROGRAMMING TUTORIAL:	22
6 - AVAILABLE KEYBOARDS	32

MagicCalc 4.49 Product Manual

New Features:

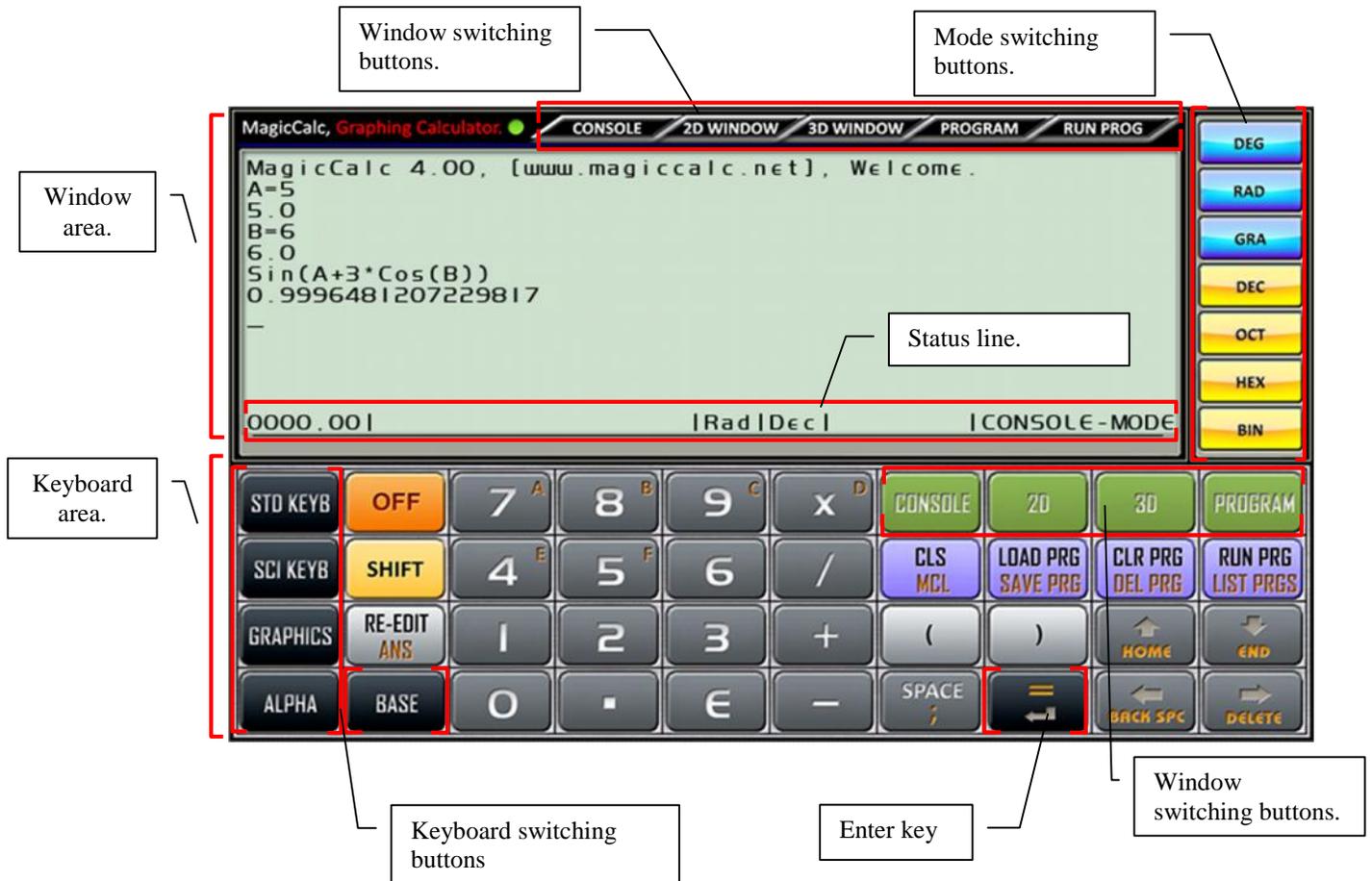
- Readln Function (In scientific keyboard), see page 12, 15 for more details.
- e (euler) constant (In scientific keyboard), see page 15 for more details.

Notes:

- to switch between capital letters and normal letter use Alpha key or Alpha + Shift key
- only 0,1 digits are available in binary mode
- only 0,1,2,3,4,5,6,7 digits are available in octal mode

-- MagicCalc is a continuous development project; many updates are coming by time... --

1 – Interface description:



In MagicCalc, and for devices with Android version greater than 2.2, multitouch is enabled for zoom and pan graphics. For all versions, simple touch is activated for moving graphics.

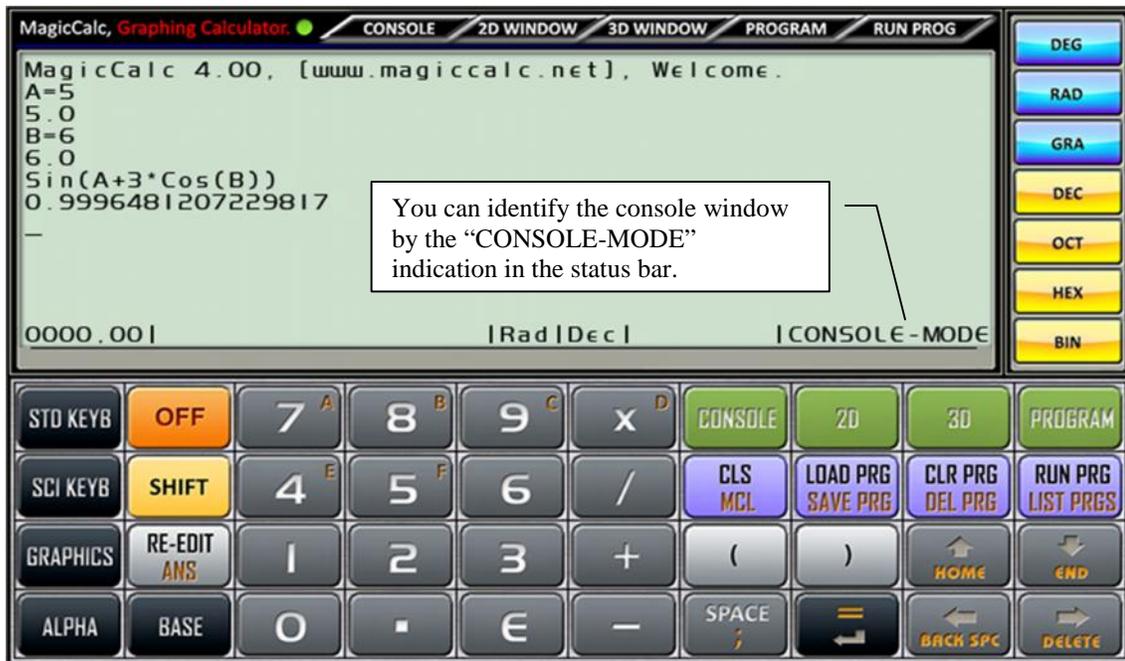


2 – Available Windows

2.1 – Switching windows:

You can switch window using the switching window buttons.

2.2 – Console window:



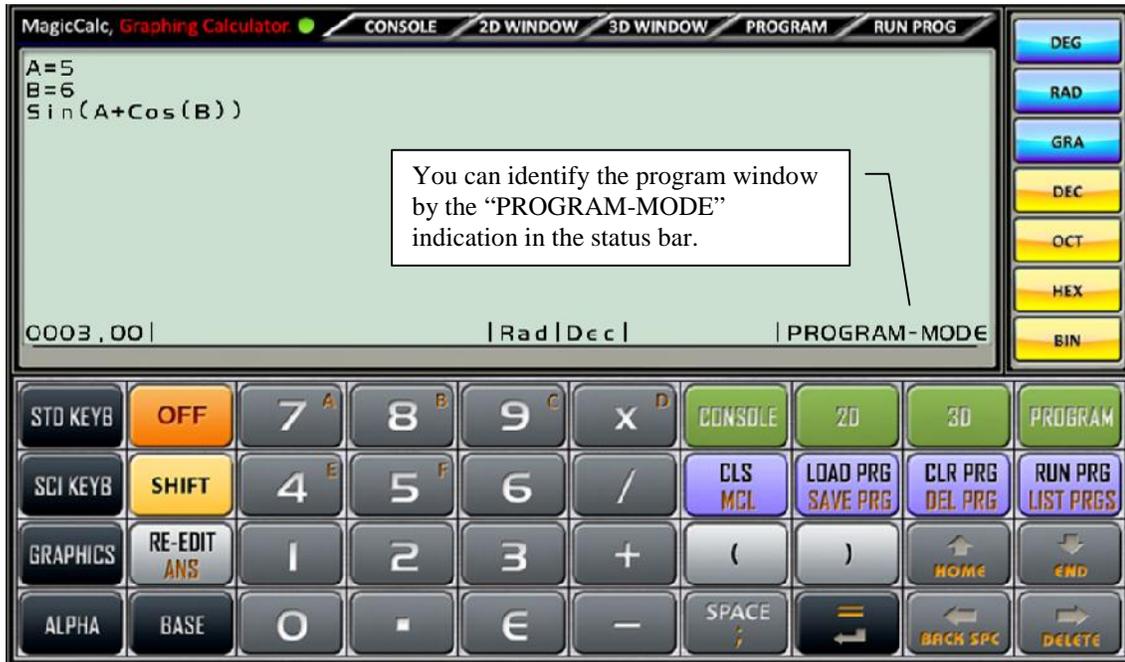
The console window is used for user entry computations. You must validate each entry by pressing the enter symbol (↵), the result appears immediately:

Example:

```
A=5 ↵
5.0
B=6 ↵
6.0
Sin(A+3*cos(B)) ↵
0.9996481207229817
```

2.3 – Program window:

2.3.1 – Presentation:



The program window is used for editing programs by users.

You can enter any sequence of instructions or computations you normally use in the console window.

Example:

$A=5$ ↵

$B=5$ ↵

$\sin(A+\cos(B))$ ↵

When you finish, click on "RUN PROGRAM" on the window switching buttons, or on "RUN PRG" on the standard keyboard, the application switches automatically to the console window, and runs the program:



Note:

- To clear the screen use the "CLS" button in the standard keyboard screen.
- To run a program, just click on Run Program in the standard keyboard.
- You can call all available functions on magiccalc.
- You can save your programs using the "SAV PRG" button on the standard keyboard.
- You can save your load your programs using the "LOAD PRG" button on the standard keyboard.
- You can clear you program using the "CLR PRG" button on the standard keyboard.
- To view the list of all saved programs, use "LIST PRGS" on the standard keyboard.

2.4 – 2D Window:**2.4.1 – 2D Window:**

Trace point.

Use the TRACE NXT and TRACE OFF buttons to trace next function, or switch trace mode off

Use these buttons in the graphics keyboard for changing the scale position, size and rotation.

Use left and right key, to move the trace over the traced function.
You can also move the trace point by passing your finger over the graph.

The 2d window is used for viewing 2d graphed functions.

2.4.2 – 2D Scale manipulation:

The scale in 2d window is by default sized to:

- X_{Min2D} : -10, X_{Max2D} : 10, X -Units : 1.0
- Y_{Min2D} : -10, Y_{Max2D} : 10, Y -Units : 1.0



However, you can redefine the 2d scale configuration & range using: X_{MIN} 2D, X_{MAX} 2D, Y_{MIN} 2D, Y_{MAX} 2D, X -UNITS and Y -UNITS buttons in the graphics keyboard:

Example:

1- Tape the following instructions in the console window:

$X_{Min2D} = -5$ ←

$X_{Max2D} = 5$ ←

$Y_{Min2D} = -5$ ←

$Y_{Max2D} = 5$ ←

2- Switch to 2D window, you will notice that the scale configuration has changed:



1- Type the following instructions in the console window:

$XUnit = 0.1$ ↵

$YUnit = 0.1$ ↵

2- Switch to 2D window, you will notice that the scale configuration has changed:

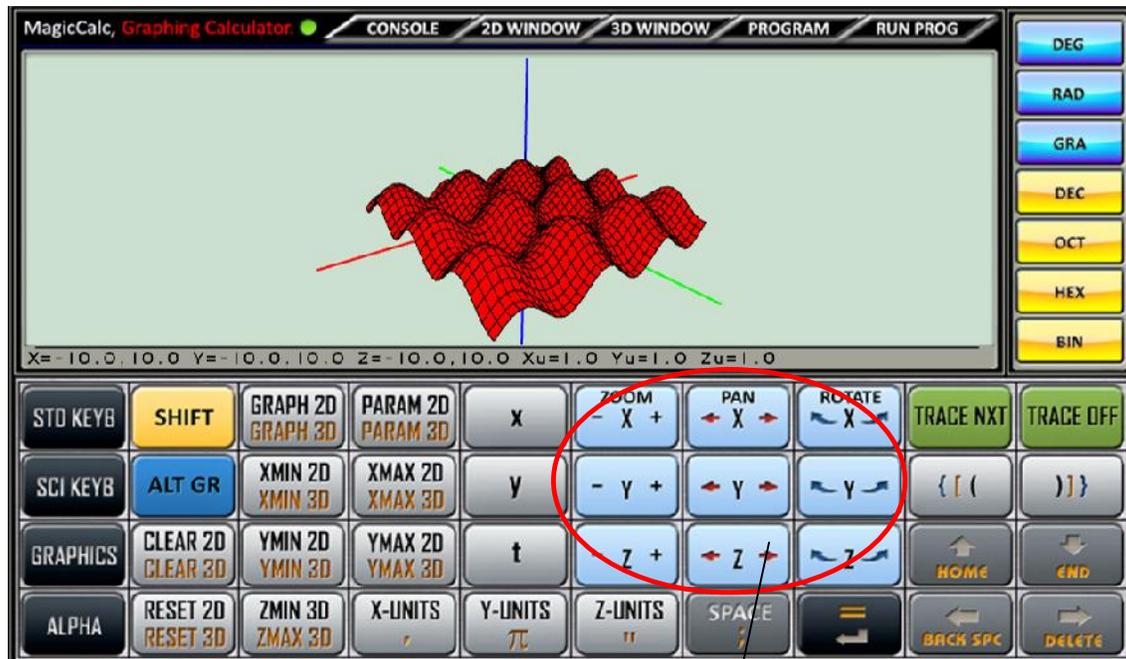


Note:

- You can reset the 2D scale by typing **Reset2D** command in the graphics keyboard.
- You can clear the 2D window by typing **Clear2D** command in the graphics keyboard.

2.5 – 3D Window:

2.5.1 – 3D Window:



Use these buttons in the graphics keyboard for changing the scale position, size and rotation.

The 3d window is used for viewing 3d graphed functions.

2.5.2 – 3D Scale manipulation:

You can configure the 3d scale window with the same philosophy of 2D scale manipulation in the previous section.

You can redefine the 3D scale using: XMIN 3D, XMAX 3D, YMIN 3D, YMAX 3D, ZMIN 3D, ZMAX 3D

Example:

1- Tape the following instructions in the console window:

$XMin3D = -15$ ↵

$XMax3D = 15$ ↵

$YMin3D = -15$ ↵

$YMax3D = 15$ ↵

$ZMin3D = -15$ ↵

$ZMax3D = 15$ ↵

2- Switch to 3D window, you will notice that the scale configuration has changed:

Note:

- You can reset the 3D scale by typing **Reset3D** command.
- You can clear the 3D window by typing **Clear3D** command.

3 – Making computations

3.1 – Making computations

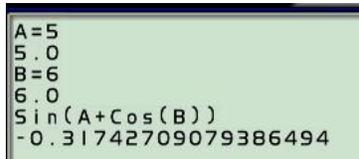
In the console window, you can enter your computations, you must validate each entry by pressing the enter symbol (↵), the result appears immediately:

Example:

```
A=5 ↵
5.0
B=6 ↵
6.0
Sin(A+3*cos(B)) ↵
0.9996481207229817
```

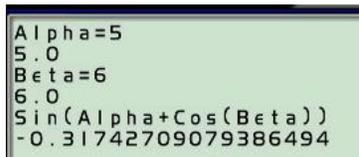
3.2 – Working with variables

You can declare any variable and use it directly on the console window or the program window.



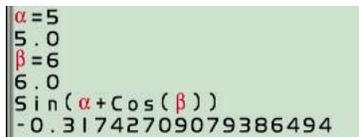
```
A=5
5.0
B=6
6.0
Sin(A+Cos(B))
-0.31742709079386494
```

You can also use complex variable names.



```
Alpha=5
5.0
Beta=6
6.0
Sin(Alpha+Cos(Beta))
-0.31742709079386494
```

You can also use Greek symbols in variables declarations.



```
α=5
5.0
β=6
6.0
Sin(α+Cos(β))
-0.31742709079386494
```

Note:

- To reset all variables and remove them from memory, use the "MCL" button in the standard keyboard screen.

3.3 – User input

You can enter variable values directly by assignment:

```
A=5 ↵
5.0
```

or using readln instruction:

```
Readln(A) ↵
```

You can also assign a variable an expression:

```
A=5 ↵
B=7 ↵
C=A+B ↵
```

3.4 – User output

You can write a string to the console using writeln instruction:

```
Writeln("Hello") ↵
```

You can query a value of a variable just by typing it's name on the console:

```
A ↵
5.0
```

You can query the value of any instruction by typing it on the console:

```
5+3 ↵
8.0
```

3.5 – Base computations

If you switch to the Base keyboard, you will have all the functions needed to work with bases.

You can switch to HEX, BIN, OCT and DEC modes using the switch modes button at the right of the top window.

Note:

- *In Binary mode, you can't use numbers other than 0 or 1. In case you do, a syntax error will be returned.*
- *In Octa Decimal mode, you can use only numbers between 0 and 7, else a syntax error will be returned.*
- *In Hexa Decimal mode, you can't use only numbers between 0 and 9, and A,B,C,D,E,F letters. In case you do, a syntax error is returned.*

3.5.1 – Bases computation keyboard:

The current computation mode is indicated on the status bar.

Or using the computation mode switch buttons at the right of the screen.

You can programmatically switch base computation mode using “BIN Mode”, “HEX Mode”, “DEC Mode” and “OCT Mode” functions.



3.5.2 – Bases logical operators

Function	Description	Example of Usage in base 2	Return
NOT	Unary Not operand for a value	NOT 1011	1111111111111111111111111111111110100
⌘	Unary Not operand for a value	⌘ 1011	1111111111111111111111111111111110100
AND	Binary and operand between 2 values	1011 AND 0101	1
OR	Binary or operand between 2 values	1011 OR 0101	1111
XOR	Binary xor operand between 2 values	1011 XOR 0101	1110
+	Addition between 2 values	1011 + 0101	10000
*	Multiplication between 2 values	1011 * 0101	110111

3.3.3 – Bases conversion functions

These function are for direct use only, users could make additions, or other computations involving theses functions.

Function	Description	Usage	Return
HEX2BIN(value)	Converts a value from hexadecimal to binary	HEX2BIN(ABC)	101010111100
HEX2OCT(value)	Converts a value from hexadecimal to octal	HEX2OCT(ABC)	5247
HEX2DEC(value)	Converts a value from hexadecimal to decimal	HEX2DEC(ABC)	2748
DEC2BIN(value)	Converts a value from decimal to binary	DEC2BIN(123)	1111011
DEC2OCT(value)	Converts a value from decimal to octal	DEC2OCT(123)	173

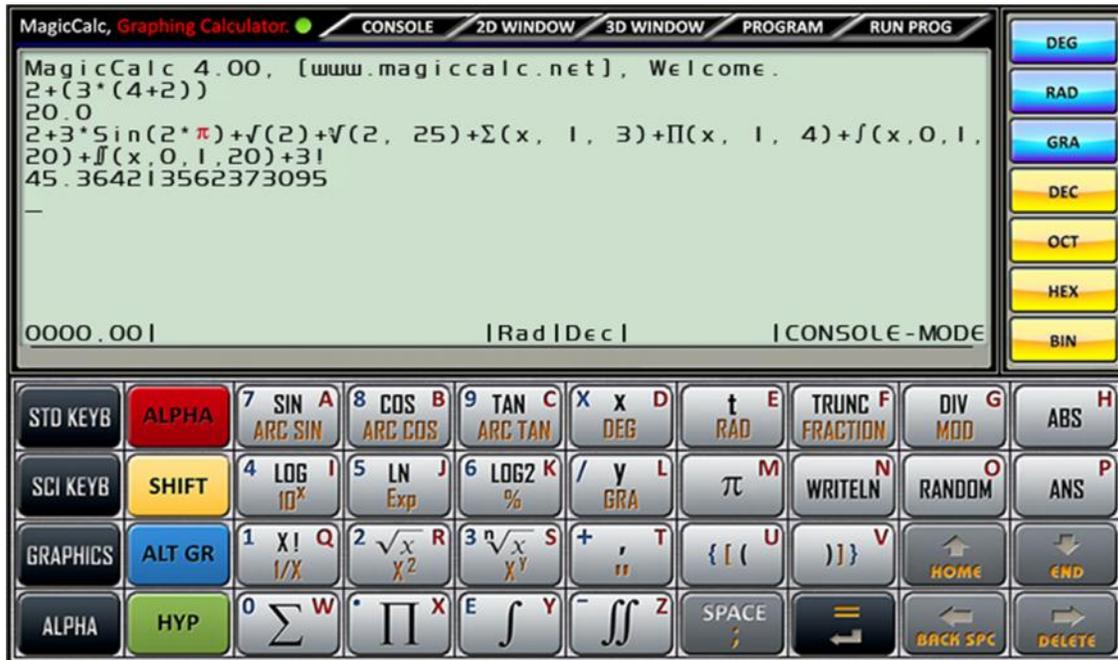
$DEC2HEX(value)$	Converts a value from decimal to hexadecimal	$DEC2HEX(123)$	7B
$OCT2BIN(value)$	Converts a value from octal to binary	$OCT2BIN(123)$	1010011
$OCT2HEX(value)$	Converts a value from octal to hexadecimal	$OCT2HEX(123)$	53
$OCT2DEC(value)$	Converts a value from octal to decimal	$OCT2DEC(123)$	83
$BIN2HEX(value)$	Converts a value from binary to hexadecimal	$BIN2HEX(10111001)$	B9
$BIN2OCT(value)$	Converts a value from binary to octal	$BIN2OCT(10111001)$	271
$BIN2DEC(value)$	Converts a value from binary to decimal	$BIN2DEC(10111001)$	185

3.5.3 – Base mode switching functions

Function	Description	Usage	Return
$BINMODE$	Switches to Binary mode	$BINMODE$	
$HEXMODE$	Switches to Hexadecimal mode	$HEXMODE$	
$DECMODE$	Switches to Decimal mode	$DECMODE$	
$OCTMODE$	Switches to Octal mode	$OCTMODE$	

3.6 – Scientific functions

The majority of the scientific functions are grouped in the scientific keyboard:



3.6.1 – Regular functions

Function	Description	Usage	Return
$Log(value)$	Return the logarithm in base 10 of the value.	$Log(16)$	1.2041199826559248
$Ln(value)$	Return the neperian logarithm of a value.	$Ln(16)$	2.772588722239781

$\text{Log2}(\text{value})$	Return the logarithm in base 2 of the value.	$\text{Log2}(16)$	4.0
$\text{Random}(\text{maxValue})$	Return a random integer between 1 and max Value .	$\text{Random}(10)$	A value between 1 and 10
$\sqrt{\text{value}}$	Return the square root of a value.	$\sqrt{16}$	4.0
$\sqrt[n]{\text{root, value}}$	Return the n^{th} root of a value	$\sqrt[2]{16}$	4.0
$\dot{y}(\text{function, minValue, maxValue})$	Return the sum of the function from min Value to Max Value .	$\dot{y}(x, 1, 3)$	6.0
$\ddot{y}(\text{function, minValue, maxValue})$	Return the product of the function from min Value to Max Value .	$\ddot{y}(x, 1, 3)$	6.0
$\int(\text{function, minValue, maxValue, precisionPoints})$	Return the Integral of a function in the interval [min Value , max Value]. The precision of the computation could be defined by precision Points .	$\int(x, 0, 1, 20)$	0.4750000000000003
$\iint(\text{function, minValue, maxValue, precisionPoints})$	Return the double Integral of a function in the interval [min Value , max Value]. The precision of the computation could be defined by precision Points .	$\iint(x, 0, 1, 20)$	0.4749999999999999
$\text{ABS}(\text{value})$	Return the absolute value of a value.	$\text{ABS}(-16)$	16.0
$\text{Trunc}(\text{value})$	Return the integer part of a real value.	$\text{Trunc}(16.23)$	16.0
$\text{Fraction}(\text{value})$	Return the decimal part of a real value.	$\text{Fraction}(16.23)$	0.23000000000000043

3.6.2 – Operators:

Function	Description	Usage	Return
div	Return the quotient of the Euclidian of 2 numbers.	$11 \text{ div } 2$	5.0
mod	Returns the remainder of the Euclidian of 2 numbers.	$11 \text{ mod } 2$	1.0
$+$	Addition operator	$2 + 3$	5.0
$-$	Substraction operator	$3 - 2$	1.0
$*$	Multiplication operator	$11 * 2$	22.0
$/$	Division operator	$8 / 2$	4.0
$^$	Power operator	$4 ^ 3$	64.0
$\%$	Percentage operator	15%	0.15
$!$	Return the factorial of a number	3!	6.0

3.6.3 – Constants:

Function	Description	Usage	Return
π	Return the value of π	π	3.141592653589793
e	Return the value of Euler constant	e	2.718281828459045

3.6.4 – Utility functions:

Function	Description	Usage	Return
Ans	Return the last result on the console	Ans	Return the last result on the console
WriteLn(string)	Write a string in a new line	WriteLn("Hello")	Hello
ReadLn(variable)	Write a string asking for that variable, and wait for user input	ReadLn(A)	A? Wait for user input Assign user input to A

3.4.5 – Trigonometric mode switching functions

Function	Description	Usage	Return
DEG MODE	Switches to Degree mode	DEG MODE	

<i>RADMODE</i>	<i>Switches to Radian mode</i>	<i>RADMODE</i>	
<i>GRAMODE</i>	<i>Switches to Gradian mode</i>	<i>GRAMODE</i>	

3.6.6 – Trigonometric functions:

<i>Function</i>	<i>Description</i>	<i>Example of usage in radian mode</i>	<i>Return</i>
<i>Sin</i>	<i>Return the sine of a value</i>	<i>Sin(1.14)</i>	<i>0.9086334961158832</i>
<i>Cos</i>	<i>Return the cosine of a value</i>	<i>Cos(1.14)</i>	<i>0.4175945039583582</i>
<i>Tan</i>	<i>Return the tangent of a value</i>	<i>Tan(1.14)</i>	<i>2.1758751312648754</i>

3.6.7 – Inverse trigonometric functions:

Using the “SHIFT” key in the scientific keyboard, you can access these functions.

<i>Function</i>	<i>Description</i>	<i>Example of usage in radian mode</i>	<i>Return</i>
<i>ArcSin</i>	<i>Return the arc sine of a value</i>	<i>ArcSin(0.9)</i>	<i>1.1197695149986342</i>
<i>ArcCos</i>	<i>Return the arc cosine of a value</i>	<i>ArcCos(0.9)</i>	<i>0.45102681179626236</i>
<i>ArcTan</i>	<i>Return the arc tangent of a value</i>	<i>ArcTan(0.9)</i>	<i>0.8507256330207998</i>

3.6.8 – Hyperbolic trigonometric functions:

Using the “HYP” key in the scientific keyboard you can access these functions.

<i>Function</i>	<i>Description</i>	<i>Example of usage in radian mode</i>	<i>Return</i>
<i>SinHyp</i>	<i>Return the hyperbolic sine of a value</i>	<i>SinHyp(0.9)</i>	<i>1.0265167257081753</i>
<i>CosHyp</i>	<i>Return the hyperbolic cosine of a value</i>	<i>CosHyp(0.9)</i>	<i>1.4330863854487745</i>
<i>TanHyp</i>	<i>Return the hyperbolic tangent of a value</i>	<i>TanHyp(0.9)</i>	<i>0.7162978701990245</i>

3.6.9 – Inverse hyperbolic trigonometric functions:

Using the “HYP” + “SHIFT” keys in the scientific keyboard you can access these functions.

<i>Function</i>	<i>Description</i>	<i>Example of usage in radian mode</i>	<i>Return</i>
<i>ArcSinHyp</i>	<i>Return the arc hyperbolic sine of a value</i>	<i>ArcSinHyp(0.9)</i>	<i>0.8088669356527826</i>
<i>ArcCosHyp</i>	<i>Return the arc hyperbolic cosine of a value</i>	<i>ArcCosHyp(1.2)</i>	<i>0.6223625037147785</i>
<i>ArcTanHyp</i>	<i>Return the arc hyperbolic tangent of a value</i>	<i>ArcTanHyp(0.9)</i>	<i>1.4722194895832201</i>

4 – Making graphics

4.1 – Presentation

The graphing system supports until 10 2d functions, and 10 3d functions. In the console window, you can enter your graphs definition using the graphing functions, you must validate each entry by pressing the enter symbol (\leftarrow), the result appears immediately in the appropriate 2d or 3d windows.

- You can clear the 2D window by typing **Clear2D** command.
- You can reset the 2D scale by typing **Reset2D** command.
- You can clear the 3D window by typing **Clear3D** command.
- You can reset the 3D scale by typing **Reset3D** command.

4.2 – 2D functions

Syntax: **Graph2D (function(x))**

Example:

Type **Graph2D** with the function you need:

Clear2D \leftarrow

XMin2D = -2.26 \leftarrow

XMax2D = 1.34 \leftarrow

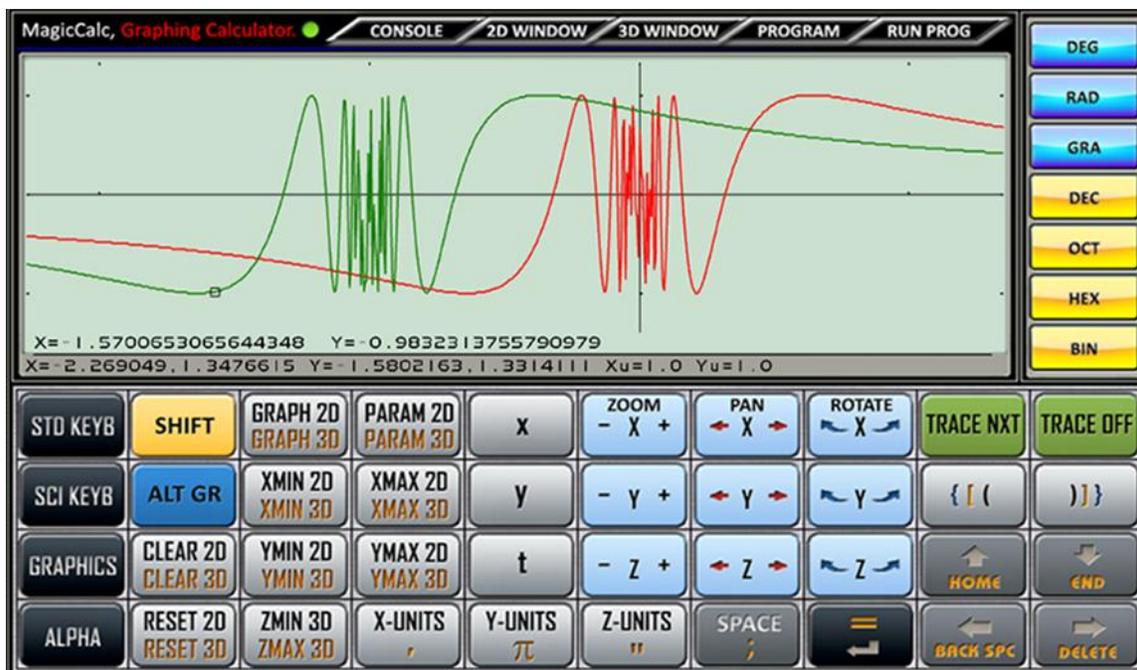
YMin2D = -1.56 \leftarrow

YMax2D = 1.34 \leftarrow

Graph2D(sin(1/x)) \leftarrow

Graph2D(sin(1/(1+x))) \leftarrow

The function will appear in the 2D Window.



4.3 – 2D parametric functions:

Syntax: **Param2D**(function1(t), function2(t), Number of points, Precision steps)

Example:

Type **Param2D** with the function you need:

Clear2D↵

XMin2D = -2.82↵

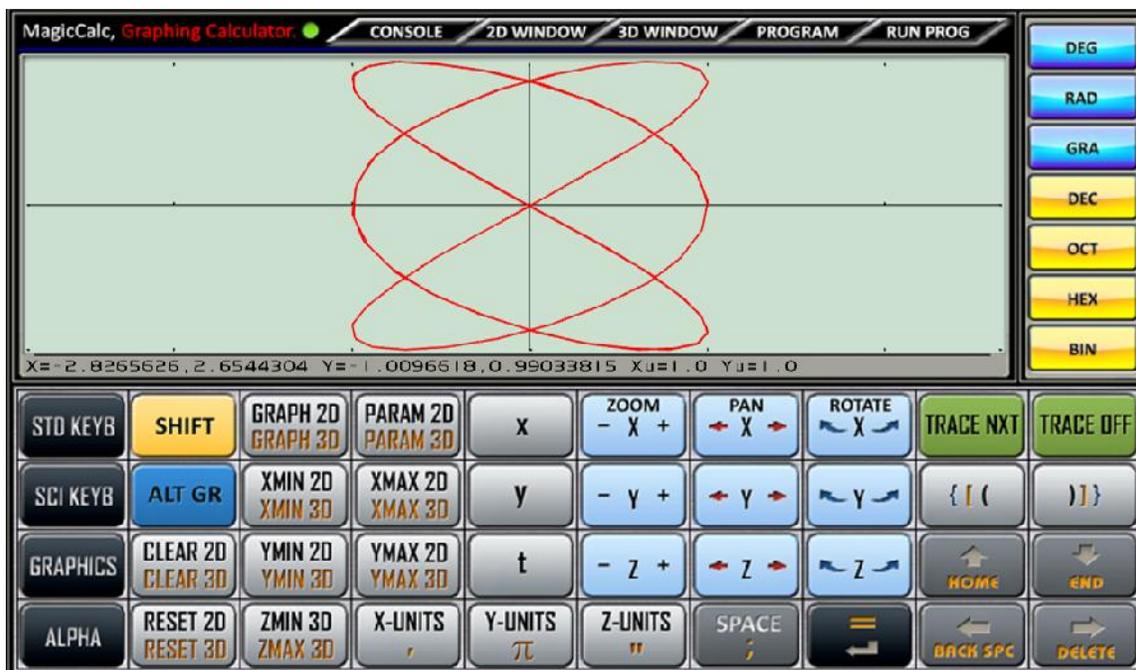
XMax2D = 2.65↵

YMin2D = -1↵

YMax2D = 1↵

Param2D($\cos(3*t)$, $\sin(2*t)$, 100, 0.1)

The function will appear in the 2D Window.



4.4 – 3D functions:

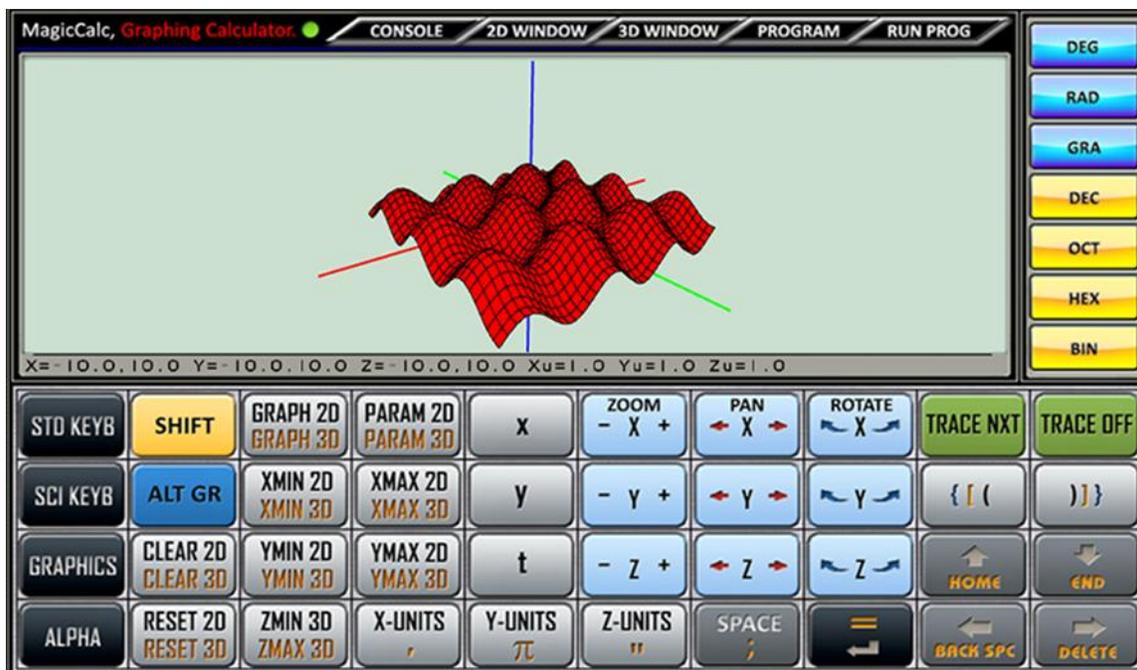
Syntax: **Graph3D** (function(x,y))

Example:

Type *Graph3D* with the function you need:

$$\text{Graph3D}(\sin(x)+\cos(y))$$

The function will appear in the 3D Window.



4.5 – 3D parametric functions:

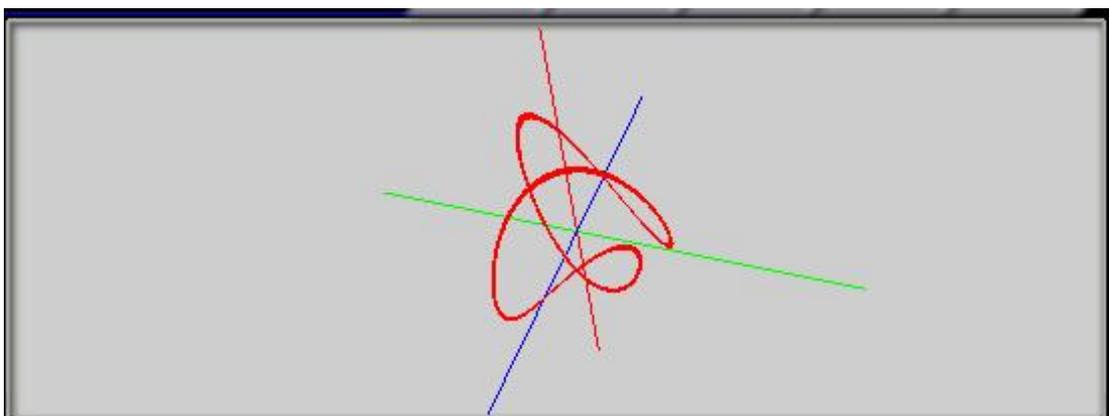
Syntax: **Param3D(function1(t), function2(t), function3(t), Number of points, Precision steps)**

Example:

Type *Param3D* with the function you need:

$$\text{Param3D}(5*\cos(3*t), 5*\sin(2*t), 5*\sin(t), 100, 0.1)$$

The function will appear in the 3D Window.



4.6 – Graphing limitations:

These functions could not be used inside graphing functions:

<i>Function</i>	<i>Description</i>
$\sum(\text{function}, \text{minValue}, \text{maxValue})$	<i>Return the sum of the function from minValue to MaxValue.</i>
$\prod(\text{function}, \text{minValue}, \text{maxValue})$	<i>Return the product of the function from minValue to MaxValue.</i>
$\int(\text{function}, \text{minValue}, \text{maxValue}, \text{precisionPoints})$	<i>Return the Integral of a function in the interval [minValue, maxValue]. The precision of the computation could be defined by precisionPoints.</i>
$\iint(\text{function}, \text{minValue}, \text{maxValue}, \text{precisionPoints})$	<i>Return the double Integral of a function in the interval [minValue, maxValue]. The precision of the computation could be defined by precisionPoints.</i>

5 - Programming tutorial:

Example 1:

1 – In the Program Window, Type the following program:

```

Cls↵
Clear2D↵
Reset2D↵

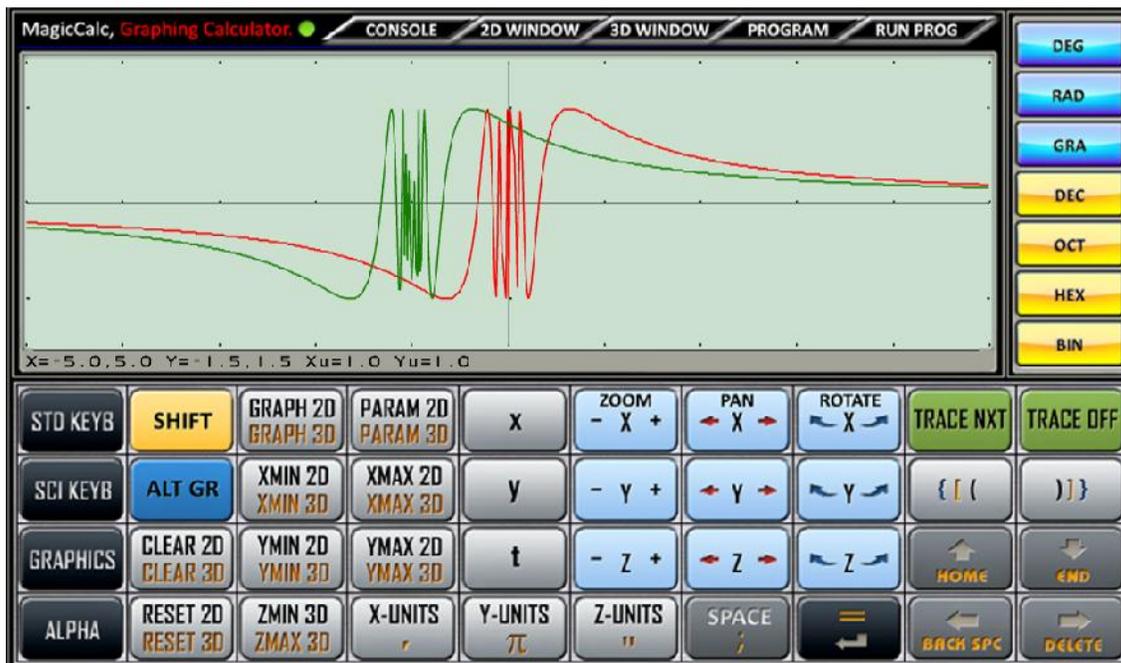
XMin2D = -5↵
XMax2D = 5↵
YMin2D = -1.5↵
YMax2D = 1.5↵

Graph2D(sin(1/x))↵
Graph2D(sin(1/(1+x)))↵

```

2 – Hit run program button

Result:



Example 2:

1 – In the Program Window, Type the following program:

Cls↵

Mcl↵

Writeln("temperature in Celcius");

C=25↵

Writeln("temperature in Fehrenheit");

C * 1.8000 + 32.0↵

2 – Hit run program button

Result:



Example 3:

1 – In the Program Window, Type the following program:

$X_{Min2D} = -1.5$ ↵

$X_{Max2D} = 1.5$ ↵

$Y_{Min2D} = -1$ ↵

$Y_{Max2D} = 1$ ↵

$Param2D(cos(A*t), sin(B*t), 100, 0.1)$ ↵

2 – In the Console window Type:

Cls↵

Clear2D↵

Reset2D↵

$A=3$ ↵

$B=3$ ↵

3 – Hit Run Program Button

Result:

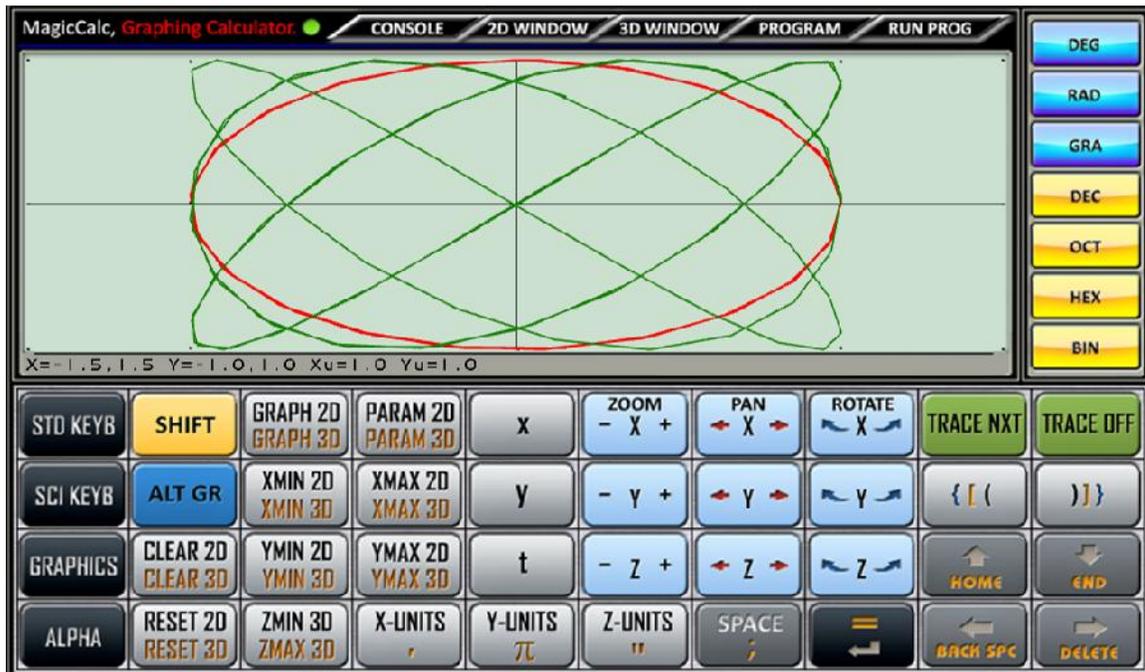


4 - In the Console window Type:

$B=4$ ↵

5 - Hit Run Program Button

Result:

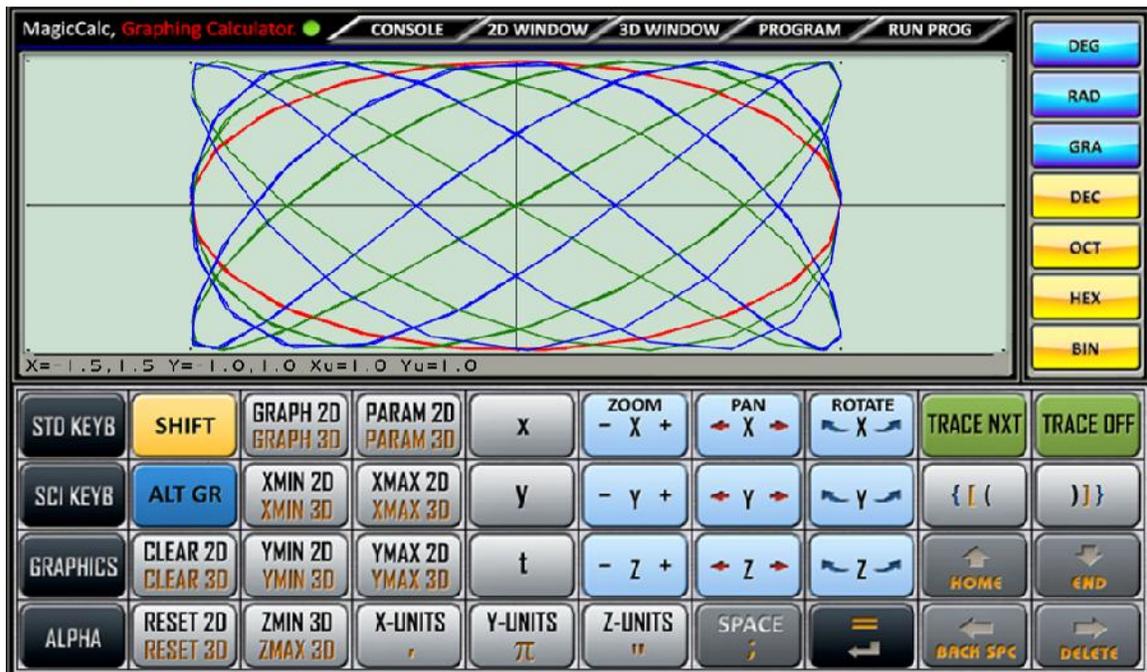


4 - In Console Mode Type:

$B=5$ ↵

5 - Hit Run Program Button

Result:



Example 4:

1 – In the Program Window, Type the following program:

```

BINMODE↵
Writeln("In binary mode, 1011+0100=");
1011+0100↵

DECMODE↵
Writeln("In decimal mode, 125+25=");
125+25↵

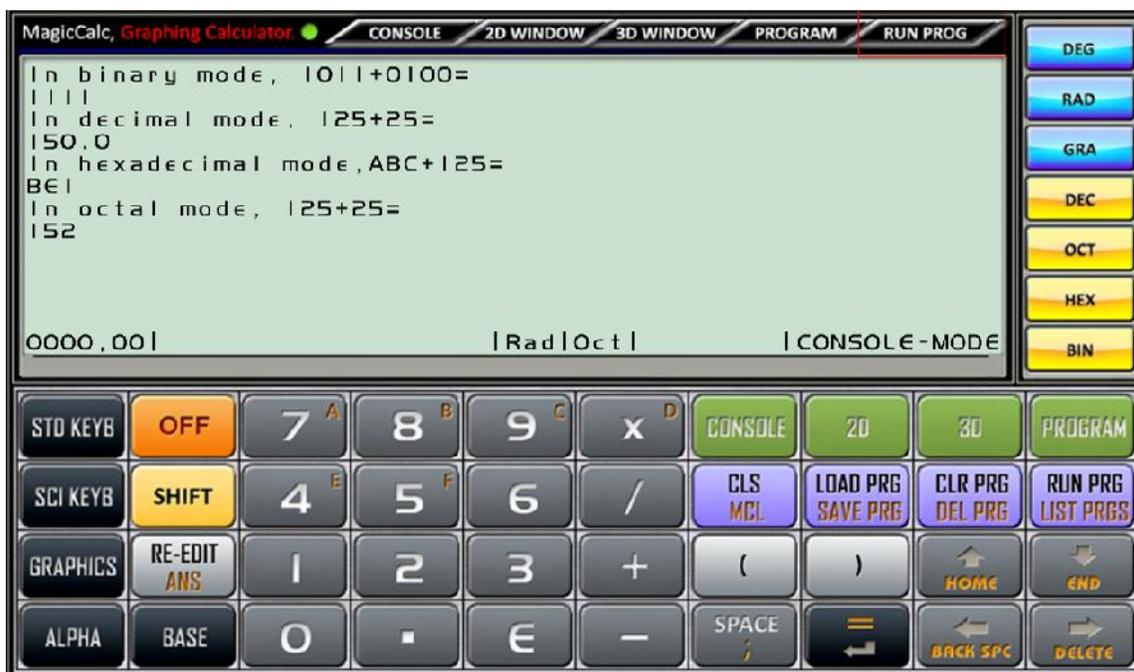
HEXMODE↵
Writeln("In hexadecimal mode, ABC+125=");
ABC+125↵

OCTMODE↵
Writeln("In octal mode, 125+25=");
125+25↵

```

2 – Hit run program button

Result:



Example 5:

1 – In the Program Window, Type the following program:

Clear2D↵

XMin2D=-100↵

XMax2D=100↵

YMin2D=-1↵

YMax2D=1↵

RADMODE↵

Graph2D("sin(x)");

DEGMODE↵

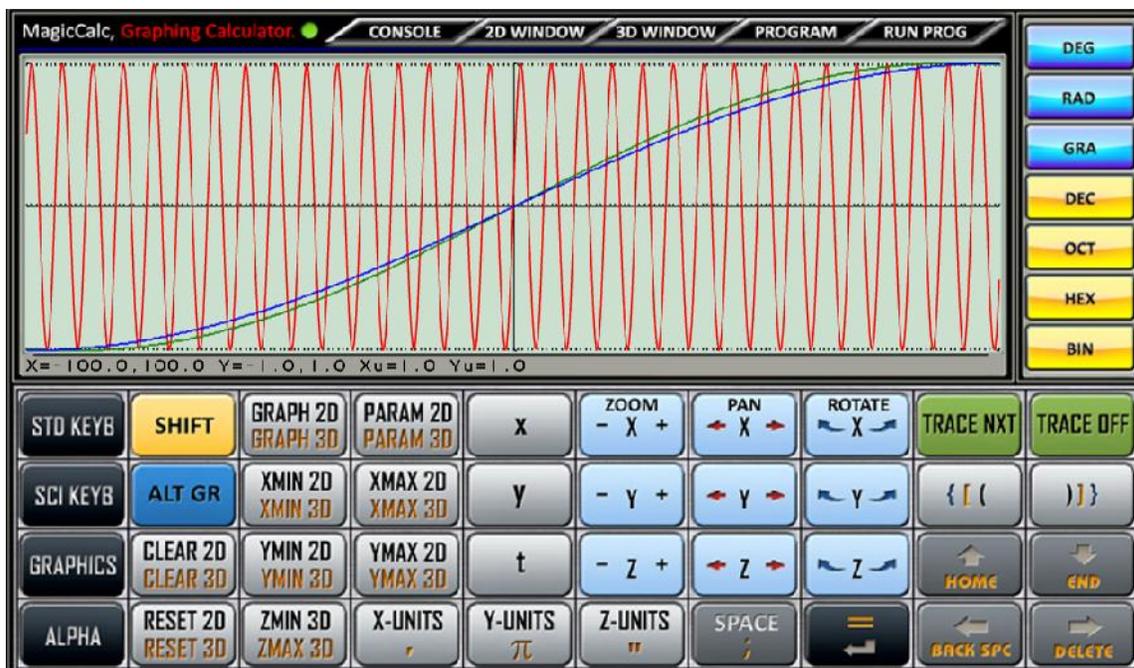
Graph2D("sin(x)"); ↵

GRAMODE↵

Graph2D("sin(x)"); ↵

2 – Hit run program button

Result:



Example 6: Using User Input

1 – In the Program Window, Type the following program:

```
Cls↵
```

```
Mcl↵
```

```
Writeln("temperature in Celcius");
```

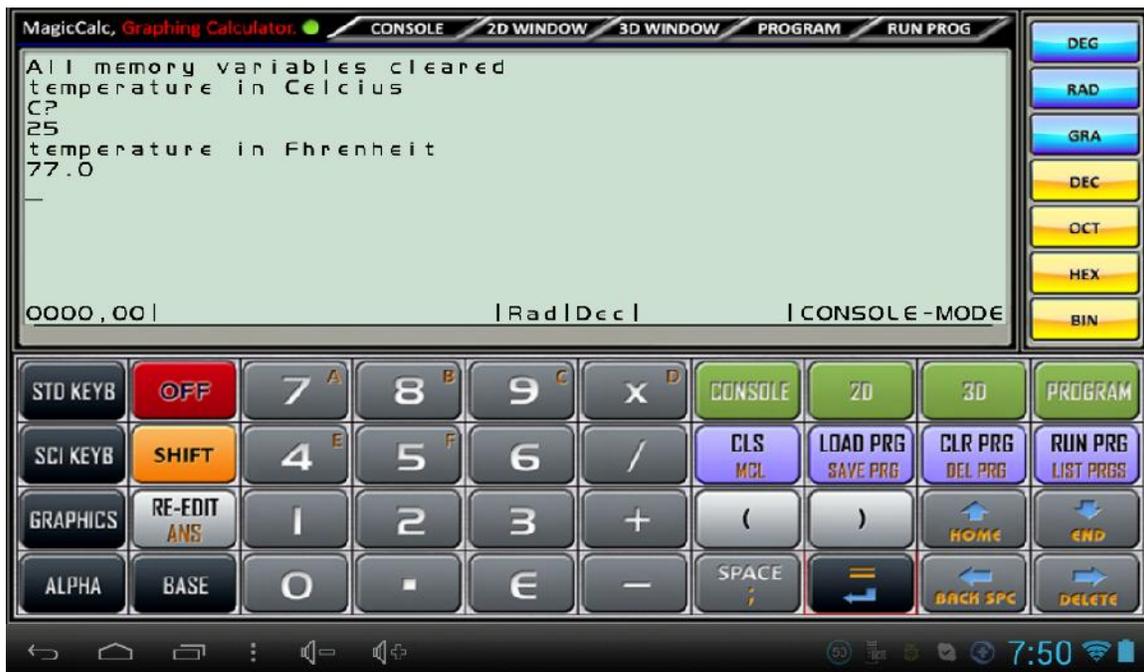
```
Readln(C)↵
```

```
Writeln("temperature in Fehrenheit");
```

```
C * 1.8000 + 32.0↵
```

2 – Hit run program button

Result:



Example 7: Using User Input

1 – In the Program Window, Type the following program:

```

Cls↵
Clear2D↵↵
Reset2D↵↵

XMin2D = -5↵↵
XMax2D = 5↵↵
YMin2D = -1.5↵↵
YMax2D = 1.5↵↵

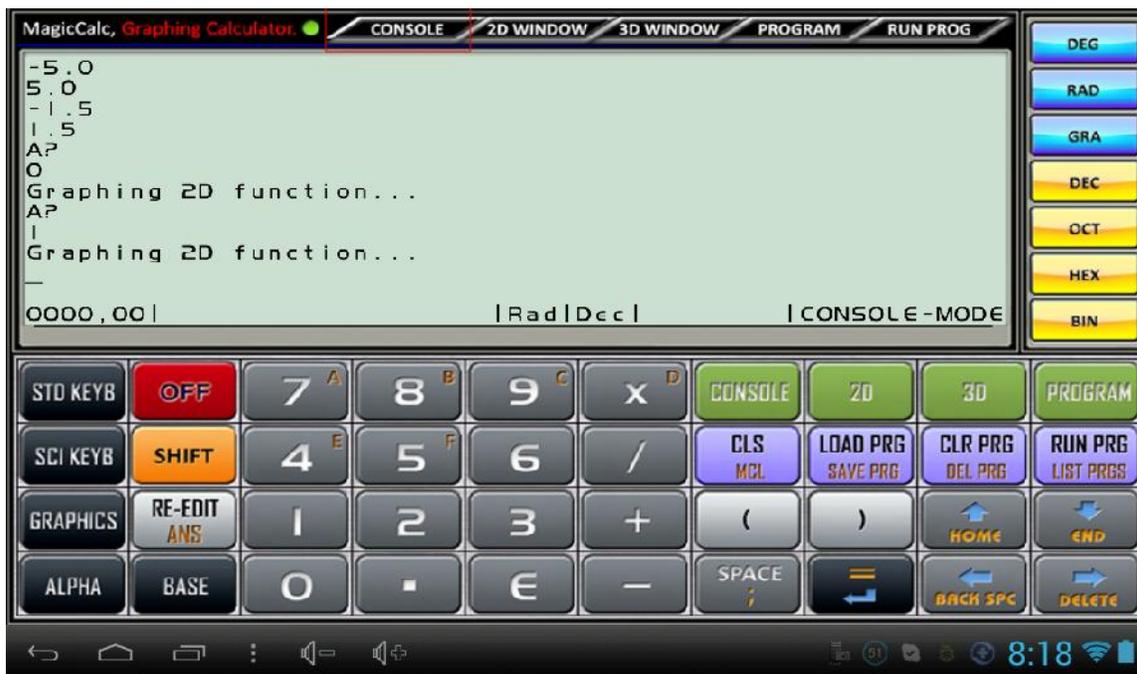
Readln(A)
Graph2D(sin(1/(x+A)))↵↵

Readln(A)
Graph2D(sin(1/(x+A)))↵↵

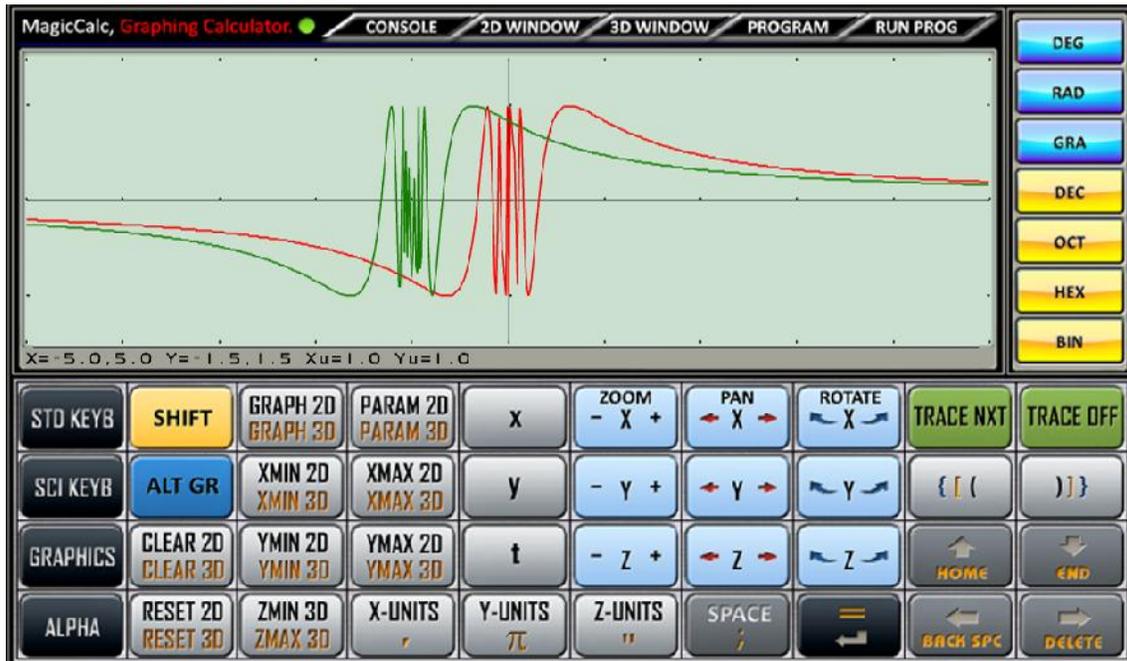
```

2 – Hit run program button

3 – Enter 0 for first entry, and 1 for second entry:



Result:



6 - Available keyboards

Standard keyboard.

You can change keyboard easily using left side buttons.



Bases computation keyboard



Scientific keyboard



Graphing keyboard



Alphabetic keyboard:

You can use the colored button in the bottom side of this keyboard to change the type of alphabetical entries.



Alphabetic capital letters:

second click on Alpha button.



Accents 1:

first click on Accents button.



Accents 2:

second click on Accents button.



Greek capital letters:

second click on Greek button.



Math symbols 1:

first click on Math Button.



Math symbols 2:

second click on Math Button.

