

# Classic Calculator Emulator +

## HP9100B Code Editor Instructions



Best viewed in 1920 x 1080 screen resolution. (Minimum is 1280 x 720)

### Disclaimer

The material contained within this package is supplied without representation or warranty of any kind. The author therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this material or any part thereof.

## Menu Items

Item	Function	Shortcut
New	Open a new file	Ctrl N
Open	Open a saved file	Ctrl O
Open Last	Open the last opened file	F5
Recent Files	View up to 5 previously loaded files	
Save	Save the current file	Ctrl S
Save As	Save the current file with new name	Ctrl F
Export	Export file to 9100 file types	Ctrl E
	Files that are exported...	
	Magnetic Card File	*.mcf
	Marked Card Reader File	*.ocf
Assemble	Assemble the code file	F9
	Creates a List file after successful assembly	*.lst
View List	View the assembled file	F12
Exit	Close the program	Ctrl X

## Assembler Instructions.

Assembler instructions can be added from the keyboard or by double clicking on an instruction from the list at the right of the screen.

Labels	Maximum length = 8 characters Must have colon character at end Must be first word on a code line
--------	--

Example	<code>MyLabel: add</code>
---------	---------------------------

## Special Instructions Format.

To make it easier to write code, there are some modified instructions available. These special instructions will be expanded to the proper 9100B instruction format when assembled.

## Comments

Comments can be added to any line and are followed by `//` characters.

Example	<code>Label1: Subtract // Subtract a value</code>
---------	---

## Set PC Address

<code>Org {+}{-}{M . B}</code>	P = Memory Address	B = Memory Nibble
--------------------------------	--------------------	-------------------

Examples	<code>Org d.0</code> <code>Org +1.0</code> <code>Org -a.d</code>
----------	--

Memory access Instructions	New Format
----------------------------	------------

<code>X&lt;-[]</code>	<code>X&lt;-[m]</code>
-----------------------	------------------------

<code>X-&gt;[]</code>	<code>X-&gt;[m]</code>
-----------------------	------------------------

<code>Y-&gt;[]</code>	<code>Y-&gt;[m]</code>
-----------------------	------------------------

<code>Y&lt;-&gt;[]</code>	<code>Y&lt;-&gt;[m]</code>
---------------------------	----------------------------

Where [m] = 0 to 9, or a, b, c or d.

Example	X to memory a	<code>X-&gt;[a]</code>
---------	---------------	------------------------

Goto	<code>Goto {Label}</code>
------	---------------------------

Example	<code>Goto MyLabel</code>
---------	---------------------------

Gosub	<code>Gosub {Label}</code>
-------	----------------------------

Example	<code>Gosub MyLabel</code>
---------	----------------------------

## Trig Instructions

HYP, ARC, SINx, COSx, TANx

These instructions can be used as in the following examples...

SINx	COSx	TANx
Arc SINx	Arc COSx	Arc TANx
Hyp SINx	Hyp COSx	Hyp TANx
Hyp Arc SINx	Hyp Arc COSx	Hyp Arc TANx

## Number Entry

Numbers can be entered individually on each line, or as complete numbers.

### Examples

5  
2  
3  
4  
.  
4  
5

Or

5234.45

-67e34

876.3e-02

+25E+02

## Code Example

This code will plot the function  $F(x)$  onto the plotter simulation accessory.

```
Clear

// store Plotter translation constants
Pi          // X Shift
Chg Sign
Shift Up
6
Multiply
Y->[c]
-.3         // Y Shift
X->[d]

15          // n
Shift Up
.15        // Y Scale
Shift Up
Pi         // X Scale

// calculate and store deltaX = Xscale / n
Roll Up
Divide
Y->[b]
Multiply

// Calculate and store Xscale/500 and 500/YScale
500
Divide
Y->[a]
Roll Up
Divide
b
Y->[b]
Shift Up
c
x<->y
Subtract

// Initialise X to Yshift - deltaX and store
ACC+
PlotLoop: RCL
Add
Y->[e]
Shift Down

// subroutine calculate f(x)
Gosub FofX // do f(x)

// calculate Yplot = 500/Yscale x (y - Yshift)
d
Subtract
b
Multiply
```

```

// calculate Xplot = 500/Xscale x (x - Xshift)
e
Shift Up
c
Subtract
a
Divide

// plot the calculated line segment
Shift Down
FMT
Shift Down
GoTo PlotLoop    // continue

FofX: Shift Up
SINx
x<->y
Divide
sub return
End

```

#### Part of the above List File after assembly...

+0.0	20	Clear	
+0.1	56	Pi	// X Shift
+0.2	32	Chg Sign	
+0.3	27	Shift Up	
+0.4	06	6	
+0.5	36	Multiply	
+0.6	40	Y->[]	
+0.7	16	c	
+0.8	21	.	// Y Shift
+0.9	03	3	
+0.a	32	Chg Sign	
+0.b	23	X->[]	
+0.c	17	d	
+0.d	01	1	// n
+1.0	05	5	
+1.1	27	Shift Up	