

RPNOS

for the TI83/84 plus

Why?

This program was produced out of my desire to have an R.P.N. calculator, while my status as a poor, impoverished student meant that I lacked the funds to buy one.

How?

The first version was written in TIBasic, however it proved to be too slow, causing an increasing delay between key presses as the number of functions increased.

Brushing the thick layers of dust from my knowledge of Z80 assembly language which have sat stagnating since the latter part of the 80s, I set about a machine code version.

The whole process was made much easier with the use of Jeremy Goetsch's superb assembler, 'Assembly Studio 8x', available from:

<http://www.ticalc.org/archives/files/fileinfo/158/15892.html>

What does it do ?

The functions currently supported are listed below:

$+$, $-$, \times , \div - *of course*

Sin , Arcsin

Cos , Arccos

Tan , Arctan

Reciprocal (x^{-1})

$\sqrt{}$, x^2

Log , 10^x

Ln , e^x

$^{}$ (*indices*)

Factorial (!) - *on the comma key due to the lack of anything more appropriate.*

L.C.M. -*on the ' (' key for the same reason as above.*

H.C.F. -*on ') ' , ditto*

nCr -*on the 'C' key*

nPr -*on the 'P' key*

Being somewhat essential, the ' π ' (pi) key also works.

The ' (-) ' key toggles the sign of the current input.

RPN specifics:

The '**Clear**' key will either clear the current input line or 'DROP' the entry at the bottom of the stack.

RPNOS uses an 8 level stack, which fits nicely into the calculator screen. Exceeding this limit will remove the top item from the stack.

The '↑' and '↓' keys 'ROLL' the stack up or down.

The '→' key functions as 'DUP', duplicating the bottom item on the stack.

The '←' key functions as 'SWAP', swapping the two bottom values on the stack.

'Enter' does exactly what you would expect.

How accurate is it?

All results are stored and calculated with a comfortable 14 digit mantissa and an exponent of up to ± 127 .

Also see '*What happens if it throws a wobbly?*' below.

So how do I make it work ?

From '**Catalog**', select '**Asm** (', then press '**PRGM**' and select '**RPN**' from the program menu . Then press '**Enter**' .

This will bring up the splash screen, where pressing any key will start the program.

And how do I get out again?

This is really quite important.

To exit from RPNOS **always** do so via the '**Quit**' ('^{2nd} ', '**Mode**') button.

Using ' Off ' will close the program, but this method causes a nasty memory leak which will quickly use up your calculator's limited RAM.

All this and programmable too ?

Ok, so this feature isn't great and was written more as an intellectual exercise than an high-powered killer app. But it works!

In 1977 my father bought one of the new-fangled TI-58 calculators which had the facility to record a series of key strokes. I was seriously impressed by this at the time, although my Sinclair Scientific couldn't be entirely discarded as it used RPN. And then there were those glorious purple LEDs. This approach to programmability is the one that I have taken, albeit to a more limited degree.

RPNOS will store a series of 95 key strokes in each of 4 program slots. This may not seem much – the TI-58 held many more all those years ago – but has so far proved more than enough for me. Should the need arise it is a small matter to increase the capacity.

To record a program, press the '**APPS**' key, which will bring up the recording menu. Pressing any key other than '1' – '4' will quit this process.

You will then be asked to enter the four letters of the program name, followed by 'Enter' .
 RPNOS will record all your key presses until either you press 'APPS' again, an error occurs or the limit of 95 keys is reached. A message will be displayed informing you that recording has finished, and pressing any key will return you to the input sequence.
 The program can then be run using the 'PRGM' key which will take you to the program selection menu. Once again, pressing any key except '1' – '4' will exit the menu.

Obviously, in order to make this feature useful, some dummy arguments should be used. As an example, the following trivial program will perform the calculation:

$a \ b \ c \ + \ x$ or $a(b+c)$ in infix notation

where c is the bottom number on the stack, with b then a above it.

1 , Enter	<i>These are the dummy values, which can be</i>
2 , Enter	<i>anything you wish, so long as they don't</i>
3	<i>cause an error in the calculation.</i>
APPS	
1	
D , E , M , O , Enter	
+	
x	
APPS	

If you were now to put the values 7, 6 and 13 on the stack then pressing:

PRGM
1

would leave an answer of 133 on the stack.

A screen capture of this process is included in this folder as an animated GIF called 'DEMO.GIF'.

The other GIF in this folder shows the full process from starting RPNOS through to entering and using a program to calculate the real roots of a quadratic equation : $(2x^2 + 3x + 1)$. It uses the dummy values 1, 2 and 1 as the coefficients. The full listing of this program is given below.

```

SWAP , (-) , 1 , x , DUP , DUP , x2
↑ , ↑ , SWAP , DUP , 2 , x , ↓
4 , x , x , - , √ , DUP , ↓ , + , SWAP
↑ , - , ↑ , DUP , ↓ , ÷ , SWAP , ↑ , ÷

```

This may seem a little long-winded, but it can be a considerable chore deciding where the brackets should be when entering using infix notation.

This is all well and good, but I don't want to enter that every time I turn my calculator on.

Never fear!

Whenever the program is exited correctly – (*see above*) – the stack and all programs are saved exactly as you left them.

What happens if it throws a wobbly?

RPNOS is fully error-trapped and the table below gives the valid input ranges for the supported functions. If an error message appears then pressing any key will return you to the input sequence. The stack will remain intact.

As mentioned above, if an error occurs while entering a program the recording process will cease.

Function	Input Range
Sin x , Cos x , Tan x	$0 \leq x < 10^{12}$ radians or degrees
Arcsin x , Arccos x	$ x < 1$
Log x , Ln x	$10^{-100} < x < 10^{100}$
e^x	$-10^{100} < x \leq 230.25850929940$
10^x	$-10^{100} < x < 100$
\sqrt{x}	$0 \leq x < 10100$
$x!$	$0 \leq x \leq 69$

Doesn't this leave a lot of the buttons on my calculator rather with nothing to do?

Yes!

However, if I find that there is functionality that I require then it will be added as and when.

Kian Vincent

February 2011