

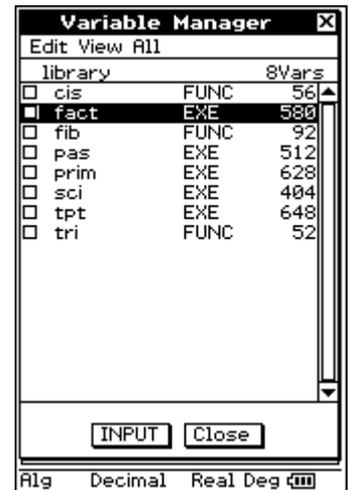
<b>Classpad Help Series sponsored by Casio Education Australia</b>		<b>www.casioed.net.au</b>	
<b>984</b>	<b>Example Small Programs</b>	Author	Charlie Watson
		Date	18 March 2011
		CPM OS	<b>03.05.0000</b>

Here are some examples of small programs to add further functionality to your ClassPad. These mini-programs can be created on your hand held or downloaded from the internet and sent to your handheld using the USB cable and free software that came with your Classpad.

Placing these mini-programs into the Other Data/Library folder on your Classpad allows access to them not only from Main and the Program application but also from within any eActivity.

*Note: In most cases, Classpad must be running OS 3.03 or later.*

An example download site is the Classpad Library section of [www.charliewatson.com/classpad](http://www.charliewatson.com/classpad)

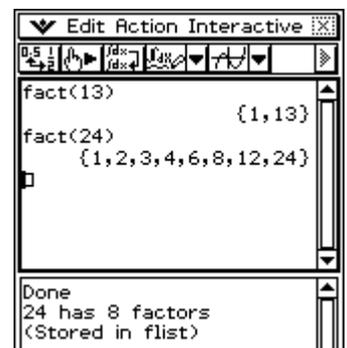


### Factors of n.

**fact** creates a list of all the factors from **1** to **n** of any positive integer **n**.

In Main enter **fact(n)**, where **n** is the number you want the factors of and they are returned in a list.

This list is stored as the variable *flist* and can be retrieved at any time by typing *flist*.

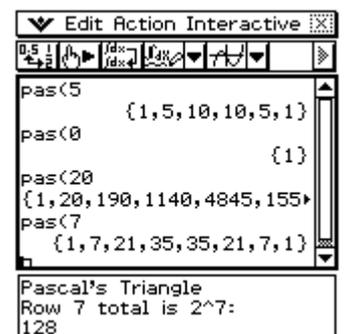


### nth row of Pascal's triangle.

**pas** creates a list of all the coefficients in the **nth** row of Pascal's triangle.

In Main, enter **pas(n)**, where **n** is the row you want and the coefficients are returned in a list.

This list is stored as the variable *plist* and can be retrieved at any time by typing *plist*.

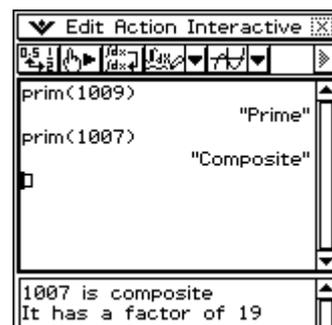


## Is n prime?

**prim** checks to see whether any number **n** is prime.

In Main enter **prim(n)**, where **n** is the number you want to check.

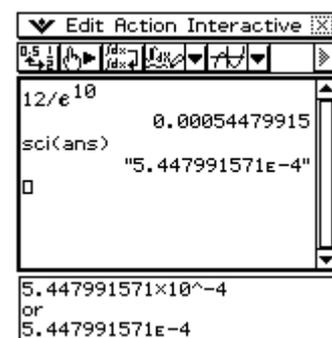
*Note: Classpad must be running OS 3.03 or later. OS 3.04 introduced the new IsPrime function, which returns TRUE or FALSE.*



## Scientific Notation.

**sci** takes any number and returns it in scientific notation (standard form).

In Main enter **sci(number)**.



## Turning point.

**tpt** takes a quadratic expression in any form and returns it in turning point (completed square) form.

In Main enter **tpt(expression)**, where expression is the quadratic you want to find the turning point of.

