HP35s / HP12c Programs

Daniel J Boulet

June 26, 2021

Revision History

Revision Date	**Description**
May 25, 2016	Revised *Modular Exponentiation* program by using DSE function instead of direct variable count.

Contents

Part I

HP35s

0.1 Modular Exponentiation

Description

This program calculates the modulus of a number raised to a large power. The formula looks like this:

$$modexp = n^p \mod m$$

Usage

`GTO A001` *n* `R/S` *p* `R/S` *m* `R/S`

Program Listing

A001	LBL A	start of program
A002	STO N	store number to the raised to the power P
A003	STOP	wait for user R/S
A004	STO P	store exponent
A005	STOP	wait for user R/S
A006	STO M	store modulus
A007	1	initialize product ...
A008	STO R	... and save in memory
A009	RCL N	recall base ...
A010	RCL R	recall product ...
A011	×	... and multiply the two
A012	RCL M	recall the modulus ...
A013	RMDR	... and apply it
A014	STO R	save the new product
A015	DSE P	decrement exponent ...
A016	GTO A009	... and loop back if not finished.
A017	RCL R	pull the product from memory
A018	RTN	we are done!

Example

In the following example we calculate $\mathbf{5^{101}}$ mod **31** using the following steps:

`GTO A001`	`0.00000`	go to start of program
`5 R/S`	`5.00000`	the "base"
`101 R/S`	`101.00000`	the "exponent"
`31 R/S`	`25.00000`	the "modulus" and result

Comments

The HP35s is not known for it's lightning speed. The above example will take about 12 seconds to run.

0.2 atan2

Description

This program calculates atan2($\frac{y}{x}$). Result is in the range $-180°$ to $+180°$.

Usage

`GTO Z001` x `R/S` y `R/S`

Program Listing

Z001	LBL Z	Start of program.
Z002	STO X	Store x in X.
Z003	STOP	Wait for user R/S.
Z004	STO Y	Store y in Y.
Z005	RCL Y	Recall Y. Note: this is also the entry point for subroutine.
Z006	RCL X	Recall X.
Z007	÷	Take ratio of rise over run ($\frac{y}{x}$).
Z008	ATAN	Calculate arctan($\frac{y}{x}$).
Z009	STO R	Save as an interim result in R.
Z010	RCL X	Test sign of X.
Z011	$x > 0$?	Is x positive?
Z012	GTO Z027	If so then go to end of program.
Z013	RCL Y	Recall Y ...
Z014	SGN	Calculate its sign ...
Z015	45	
Z016	×	then multiply it by 45°.
Z017	RCL X	Get X value.
Z018	$x = 0$?	Is it equal to zero?
Z019	RTN	If so then return the value of the stack ($\pm 45°$)
Z020	180	Setup offset depending on sign of y.
Z021	STO -R	Initially subtract 180°— we do this at a minimum.
Z022	RCL Y	Get Y value.
Z023	$x < 0$?	Is it negative?
Z024	GTO Z027	If yes, then we are done since we already subtracted 180°.
Z025	360	If y is positive then we have to add 360°...
Z026	STO +R	... for a total addition of 180°.
Z027	RCL R	Get the angle.
Z028	RTN	Return to calling function.

Example

In the following example we calculate atan2($\frac{+1.5}{-1.0}$) using the following steps:

`GTO Z001`	`0.00000`	Go to start of program.
`1.5`	`1.5`	Your value for x.
`R/S`	`1.50000`	
`-1.0`	`-1.0`	Your value for y.
`R/S`	`-33.69007`	The resulting angle.

Comments

Users have to be careful about a couple of things:

1. Angles are calculated in *degrees*. Confirm calculator setting before using this function.
2. User is responsible for ensuring that x and y are **never** both zero.