

Library 1278
Additional Functions for the HP 49G+
Version 2.07

Christopher Mark Gore

February 8, A.D. 2004

Contents

1	Introduction	5
2	Contact Information	6
3	Copyright and License	7
4	Installation Instructions	8
5	.ZIP File Contents	10
6	Version History	11
7	Function Reference	13
7.1	INTGR: Integers	13
7.1.1	FIBON	13
7.1.2	UNITSTEP	13
7.1.3	ZSEQ	14
7.1.4	RELPRIME	14
7.1.5	ZRELPRIME	14
7.1.6	ORDMULT	15
7.2	REAL: Real Numbers	15
7.2.1	ARCLEN	15
7.2.2	AVGRC	16
7.2.3	BETA	16
7.2.4	ERF	17
7.2.5	ERFC	17
7.2.6	LOGYX	17
7.2.7	LD	18
7.2.8	PFUNC	18
7.2.9	PFINV	18
7.2.10	FRESC	19
7.2.11	FRESS	19
7.3	TRIG: Trigonometry	19
7.3.1	CSC	19
7.3.2	ACSC	20

7.3.3	SEC	20
7.3.4	ASEC	20
7.3.5	COT	20
7.3.6	ACOT	21
7.3.7	VERS	21
7.3.8	COVERS	21
7.3.9	HAV	22
7.4	ANGLE: Angles	22
7.4.1	R→MODE	22
7.4.2	D→MODE	22
7.4.3	G→MODE	22
7.4.4	→R	23
7.4.5	→D	23
7.4.6	→G	23
7.4.7	R→G	23
7.4.8	D→G	24
7.4.9	G→R	24
7.4.10	G→D	24
7.4.11	ANGLE	25
7.5	HYPER: Hyperbolics	25
7.5.1	CSCH	25
7.5.2	ACSCH	25
7.5.3	SECH	26
7.5.4	ASECH	26
7.5.5	COTH	26
7.5.6	ACOTH	27
7.6	SYMB: Symbolic Manipulation	27
7.6.1	RANDPOLY	27
7.6.2	LEFT	27
7.6.3	RIGHT	28
7.6.4	GETCOEFF	28
7.6.5	GETNUM	28
7.6.6	GETDENOM	29
7.6.7	→RPN	29
7.7	PROB: Probability	29
7.7.1	IRAND	29
7.7.2	DICE	30
7.7.3	DICE6	30
7.7.4	COIN	30
7.7.5	LTPN	31
7.7.6	RNGPN	31
7.7.7	BIDIST	32
7.7.8	POIDIST	32
7.8	STAT: Statistics:	33
7.8.1	ΣMIDRANGE	33
7.8.2	Σ%TILE	33

7.8.3	Σ MEDIAN	34
7.8.4	Σ RANGE	34
7.8.5	ZSCORE	34
7.8.6	PZSCORE	35
7.8.7	Σ PF	35
7.8.8	Σ MF	35
7.8.9	%TILE	35
7.8.10	MEDIAN	36
7.8.11	QUARTILE	36
7.8.12	Q123	36
7.8.13	IQR	36
7.9	CALC: Calculus and Differential Equations	37
7.9.1	MDER	37
7.9.2	EUAPPROX	37
7.9.3	EXACTDE	38
7.9.4	WRONSKIAN	38
7.10	VECTR: Vectors	39
7.10.1	PROJ	39
7.10.2	ACROSS	39
7.10.3	BCROSS	40
7.10.4	UNITV	40
7.10.5	VBOXP	40
7.10.6	\rightarrow VEC	40
7.10.7	V \rightarrow COL	41
7.10.8	V \rightarrow V3	41
7.10.9	θ V1V2	41
7.11	MATRIX: Matrices	42
7.11.1	APLY	42
7.11.2	ADJ	42
7.11.3	GETCOL	43
7.11.4	GETROW	43
7.11.5	PUTCOL	43
7.11.6	PUTROW	44
7.12	LIST: Lists	44
7.12.1	APPEND	44
7.12.2	PREPEND	45
7.12.3	FLATLIST	45
7.12.4	LoL \rightarrow L	45
7.12.5	NGET	46
7.12.6	NUMSEQ	46
7.12.7	Σ STEP	46
7.12.8	IISTEP	47
7.13	CHARS: Characters and Strings	47
7.13.1	CHOP	47
7.13.2	LCHARS	48
7.13.3	LNUMS	48

7.13.4	NSTR	48
7.13.5	CAESAR	49
7.13.6	VIGENERE	49
7.13.7	DEVIGENERE	49
7.14	PROG: Programming	50
7.14.1	NMENU	50
7.14.2	FOLDERGROB	50
7.14.3	NCST1278	50
7.14.4	PROGCAT	50
7.14.5	NULLDIR	51
7.15	TIME: Date and Time	51
7.15.1	DAY	51
7.15.2	MONTH	51
7.15.3	YEAR	51
7.15.4	WEEKDAY	51
7.16	MISC: Miscellaneous	52
7.16.1	FWGHT	52
7.16.2	NPV	52
7.16.3	STOSTK	53
7.16.4	RCLSTK	53
8	XLIB Numbers	54

Chapter 1

Introduction

These are various functions I wrote to make my life as a engineering, and later math student easier. Many of these functions were originally written for my HP 48GX, and then later moved over to my HP 49G when that came out. It is my hope that you may find these functions useful too, saving you time and effort in your classes or work.

This library is released under a very liberal BSD-style license, excepting the functions taken from the *HP 48G AUR* [6] and the HP 48G's TEACH programs (MEDIAN and APLY), which would be HP's property. I am not sure why the TEACH program was removed with the HP 49G, but it did have a few useful functions in it.

If you find an error, please feel free to tell me of it. Various methods to contact me are listed in this document. However, if you can fix it yourself, that would be even more welcome.

Chapter 2

Contact Information

Author

Christopher Mark Gore.

World Wide Web

`<http://home.earthlink.net/~chris-gore/>`

Electronic Mail

`<chris-gore@earthlink.net>`
`<madmarlin@yahoo.com>`

Traditional Mail

Christopher Mark Gore,
8729 Lower Marine Road,
Saint Jacob, Illinois 62281.

Chapter 3

Copyright and License

Copyright 1999 – 2004 by Christopher Mark Gore. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by Christopher Mark Gore and other contributors.
4. Neither the name of Christopher Mark Gore nor the names of any other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY CHRISTOPHER MARK GORE AND OTHER CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CHRISTOPHER MARK GORE OR OTHER CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Chapter 4

Installation Instructions

This library can be installed in the normal way one installs a library for the HP 49G+, so if you already know how to install a library, you do not need to read this.

1. Transfer the library file to your HP 49G+. You would typically either use the program `conn4x` (which is freely available from Hewlett-Packard's website) or, if you have an SD card and SD card reader on your computer, copy the library to the SD card and then put the SD card into your calculator.
2. Place the calculator in RPN mode. To do this, press the MODE button, and then under Operating Mode select RPN, then press OK (F6).
3. Purge an older version of this library, if one exists on your calculator. If you have an older version of this library in port 2, for instance, you would type `2:1278 PURGE`. If you used a different port, put that number before the colon instead of 2. If you do not have an older version of this library on your calculator, do not do this.
4. Store the library to an appropriate port. Press the VAR button, and when you see the library file, press the soft key corresponding to its name. Then you should have `Library 1278: Addi...` on level 1 of your stack, this is the library. Type in the number of the port you want the library in and then press `STO`. For example, to put it in port 2 press `2 STO`. Port 0 is not recommended if there is enough free space in either port 1 or 2, since it shares memory with the rest of the system.
5. Attach the new library of functions. To attach the new library of functions, if it is stored in port 2, you would type `2:1278 ATTACH`. If it is in a port other than 2, put the appropriate port number in place of 2.
6. Reboot the calculator. Press the ON button, hold it down, and press the F3 button. Release both. The calculator should reboot now.

7. Use the functions as you wish, the library should be installed now. To access the functions, either press right-shift (the red button) LIB (the 2 button) and then the soft key corresponding to the **Addit** menu item. Once there, **Menu1** will get the library menu. You can also select the function you wish to use from the system catalog by pressing the CAT button, or by typing in the name of the function directly. You can also type in **Menu1278** to get the library's menu. Menu1278 is probably too slow to use on the original HP 49G. It takes about 3 seconds to initially load on the 49G+.

Chapter 5

.ZIP File Contents

additional_functions.dvi This documentation, in T_EX Device Independent (.DVI) format.

additional_functions.ps This documentation, in PostScript (.PS) format.

additional_functions.pdf This documentation, in Adobe Acrobat Reader (.PDF) format.

Lib1278.207 This is the actual library for use on the HP 49G+.

Lib1278K.207 This is the source code to the function library, as a Kermit-translated ASCII text file, suitable for use on a personal computer.

Lib1278S.207 This is the source code to the function library, as a HP 49G+ directory.

Chapter 6

Version History

Version 2.07, released Feb. 8, 2004: Added an improved \rightarrow RPN from Javier Goizueta.

Version 2.06, released Jan. 7, 2004: This was the first version to be released with complete documentation. Added NULLDIR, PUTCOL, and PUTROW. Enhanced GETCOL, GETROW, and PROGCAT. Minor fixes to θ V1V2, ZSCORE, and PZSCORE. Removed AFFINE, VP1P2, and Σ FREQ.

Version 2.05, released Dec. 8, 2003: This was the first 2.xx version to appear on hpcalc.org. Added PREPEND, IISTEP, \rightarrow VEC, and Σ %TILE. Minor re-writes of a few of the functions, including a 28% speed increase in NMENU. Removed CHEBYCHEV.

Version 2.04, released Nov. 30, 2003: Added WEEKDAY, LTPN, STOSTK, and RCLSTK functions. Re-wrote ERF and RNGPN for increased speed.

Version 2.03, released Nov. 26, 2003: Re-wrote and re-named a few functions to mirror related built-in functions more closely.

Version 2.02, released Nov. 22, 2003: Minor re-write of the NMENU function.

Version 2.01, released Nov. 21, 2003: Minor bug fix.

Version 2.00, released Nov. 20, 2003: Initial re-release of the function library for the 49G+.

Version 1.2, released May 18, 2000: Last version written for the 49G to be released, was never uploaded to hpcalc.org.

Version 1.1, released Mar. 12, 2000: Last version that was uploaded to hpcalc.org written for the 49G.

Version 1.0, released Feb. 9, 2000: First version to be posted on the website <http://www.hpcalc.org>.

Chapter 7

Function Reference

7.1 INTGR: Integers

7.1.1 FIBON

Fibonacci Numbers: Calculates the Fibonacci number $F(n)$, which is defined as:

$$F(n) = \begin{cases} n & n \leq 1 \\ F(n-1) + F(n-2) & n > 1 \end{cases}$$

General form:



Numerical example:



7.1.2 UNITSTEP

Unit Step: Given a number x , returns 0 if $x < 0$ and 1 if $x \geq 0$.

Numerical example:



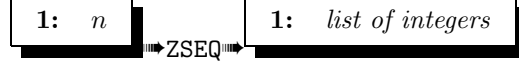
Numerical example:



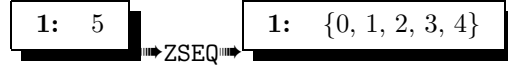
7.1.3 ZSEQ

\mathbb{Z}_n : Given positive integer n , returns the set $\{0, 1, \dots, n-2, n-1\}$, which is usually referred to as \mathbb{Z}_n .

General form:



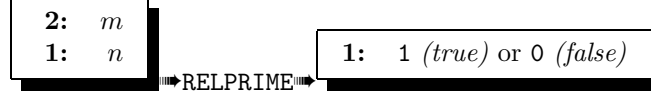
Numerical example:



7.1.4 RELPRIME

Relative Primality (Co-Primality) Test: This function, given two integers m and n , returns 1 if m is relatively prime (co-prime) to n , or 0 otherwise.

General form:



Example:



Example:



7.1.5 ZRELPRIME

\mathbb{Z}_n^* : This function, given a positive integer n , returns the set of all positive integers $i < n$ such that i is relatively prime (co-prime) to n . This set is usually represented as \mathbb{Z}_n^*

General form:



Numerical example:

1: 10	→ZRELPRIME→	1: {1, 3, 7, 9}
-------	-------------	-----------------

7.1.6 ORDMULT

Element Multiplicative Order: This function, given a positive integer n , returns the multiplicative order of the element n in the current modulus of the system, $|n|$. The current modulus may be changed by the system command MODSTO found in the **ARITH** (*left-shift 3*), **MODUL** menu.

General form:

1: n	→ZRELPRIME→	1: $ n $ (<i>in the current modulus</i>)
--------	-------------	--

Numerical example:

(*assuming modulus 13, the default*):

1: 7	→ZRELPRIME→	1: 12
------	-------------	-------

7.2 REAL: Real Numbers

7.2.1 ARCLLEN

Length of Arc: This function calculates the length of a section of a curve. This can be found by the equation:

$$\int_{start}^{end} \sqrt{1 + \left(\frac{df(x)}{dx} \right)^2} dx$$

where $f(x)$ is the function of the curve, rather easily, but the TI-92+ has it as a built-in command.

General form:

4: $start$	→ARCLLEN→	1: $arc\ length$
3: end		
2: $curve\ function$		
1: $variable\ of\ integration$		

Diagram illustrating the ARCL EN architecture. The input is a stack of elements (4: 0, 3: 5, 2: x^3 , 1: x). These elements are processed by the ARCL EN block, resulting in a single output element (1: 125.680300639).

Average Rate of Change: Given the function $f(x)$, variable x , and step size h returns the average rate of change (forward difference quotient), which is defined as:

$$\frac{f(x+h) - f(x)}{h}$$

General form

3: <i>function</i>
2: <i>variable</i>
1: <i>step size</i>

AVGRC

1: <i>forward difference quotient</i>

Complete Beta: Computes the complete Beta of p and q . The complete beta function can be defined as:

$$B(p, q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$$

```

graph LR
    A["2: p  
1: q"] -- BETA --> B["1: B(p, q)"]
  
```

2: 1.2
1: 2.35

→BETA→ 1: 0.314411531794

7.2.4 ERF

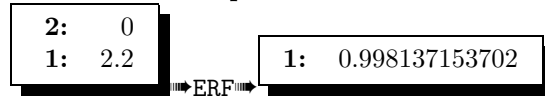
Generalized Gaussian Error: This function calculates the generalized Gaussian error function $\text{erf}(\alpha, \beta)$. In order to calculate $\text{erf}(x)$, use $\text{erf}(0, x)$. The generalized error function can be found by:

$$\text{erf}(\alpha, \beta) = \frac{2}{\sqrt{\pi}} \int_{\alpha}^{\beta} \frac{dt}{e^{t^2}}$$

General form:



Numerical example:



7.2.5 ERFC

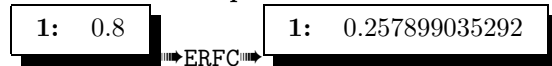
Complementary Error: Calculates the complementary error function $\text{erfc}(x)$, which is defined as:

$$\text{erfc}(x) = 1 - \text{erf}(x)$$

General form:



Numerical example:



7.2.6 LOGYX

Logarithm of Any Base: Calculates $\log_y(x)$.

General form:



Numerical example:



7.2.7 LD

Logarithm of Base 2: Calculates $\text{ld}(x)$, which is $\log_2(x)$. This is from the Latin *logarithmus dualis*, but this function is also sometimes written as $\lg(x)$.

General form:



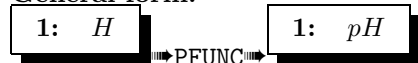
Numerical example:



7.2.8 PFUNC

Chemistry p Function: The p function is often used in chemistry and other sciences to calculate things such as pH or pOH. pH is merely defined as $-\log_{10}(H)$, but I always get it confused with its inverse function.

General form:



Numerical example:



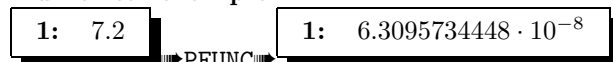
7.2.9 PFINV

Inverse Chemistry p Function: This is the inverse of the p function commonly used in chemistry and other sciences. For example, if you know the pH of something, this function will calculate the concentration of H ions in that substance. The value of H can be found merely by -10^{pH} , but I always get this and the p function confused.

General form:



Numerical example:



7.2.10 FRESC

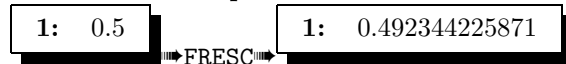
Fresnel C: Calculates the Fresnel integral $C(x)$, which is defined as:

$$C(x) = \int_0^x \cos\left(\frac{\pi t^2}{2}\right) dt$$

General form:



Numerical example:



7.2.11 FRESS

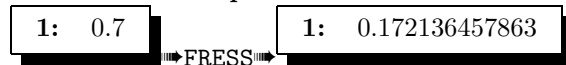
Fresnel S: Calculates the Fresnel integral $S(x)$, which is defined as:

$$S(x) = \int_0^x \sin\left(\frac{\pi t^2}{2}\right) dt$$

General form:



Numerical example:



7.3 TRIG: Trigonometry

7.3.1 CSC

Cosecant: Returns the cosecant of the argument.

General form:



Numerical example:



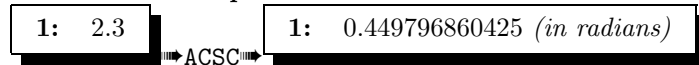
7.3.2 ACSC

Arc Cosecant: Returns the value of the angle having the given cosecant.

General form:



Numerical example:



7.3.3 SEC

Secant: Returns the secant of the argument.

General form:



Numerical example:



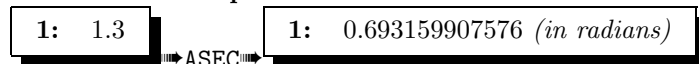
7.3.4 ASEC

Arc Secant: Returns the value of the angle having the given secant.

General form:



Numerical example:



7.3.5 COT

Cotangent: Returns the cotangent of the argument.

General form:



Numerical example:

1: 2.35 (<i>in radians</i>)	⇒COT⇒	1: -0.987687134052
-------------------------------	-------	--------------------

7.3.6 ACOT

Arc Cotangent: Returns the value of the angle having the given cotangent.

General form:

1: x	⇒ACOT⇒	1: $\text{arc cot}(x)$
--------	--------	------------------------

Numerical example:

1: 5.	⇒COT⇒	1: 0.19739555985 (<i>in radians</i>)
-------	-------	--

7.3.7 VERS

Versine: Returns the versine of the argument. The versine is defined as:

$$\text{vers}(\theta) = 1 - \cos(\theta)$$

General form:

1: θ	⇒ACOT⇒	1: $\text{vers}(\theta)$
-------------	--------	--------------------------

Numerical example:

1: 1.235 (<i>in radians</i>)	⇒VERS⇒	1: 0.670478875213
--------------------------------	--------	-------------------

7.3.8 COVERS

Coversine: Returns the coversine of the argument. The coversine is defined as:

$$\text{covers}(\theta) = 1 - \sin(\theta)$$

General form:

1: θ	⇒ACOT⇒	1: $\text{covers}(\theta)$
-------------	--------	----------------------------

Numerical example:

1: 2.2 (<i>in radians</i>)	⇒COVERS⇒	1: 0.19150359618
------------------------------	----------	------------------

7.3.9 HAV

Haversine: Returns the haversine of the argument. The haversine is defined as:

$$\text{hav}(\theta) = \frac{\text{vers}(\theta)}{2}$$

General form:



Numerical example:



7.4 ANGLE: Angles

7.4.1 R→MODE

Radians to Current Angular Mode: Converts a real number expressed in radians to its equivalent in whatever angular format the calculator is currently operating under.

General form:



7.4.2 D→MODE

Degrees to Current Angular Mode: Converts a real number expressed in degrees to its equivalent in whatever angular format the calculator is currently operating under.

General form:



7.4.3 G→MODE

Grads to Current Angular Mode: Converts a real number expressed in grads to its equivalent in whatever angular format the calculator is currently operating under.

General form:



7.4.4 $\rightarrow R$

Convert to Radians: Converts a real number expressed in whatever the current angular format the calculator is currently operating under to its equivalent in radians.

General form:



7.4.5 $\rightarrow D$

Convert to Degrees: Converts a real number expressed in whatever the current angular format the calculator is currently operating under to its equivalent in degrees.

General form:



7.4.6 $\rightarrow G$

Convert to Grads: Converts a real number expressed in whatever the current angular format the calculator is currently operating under to its equivalent in grads.

General form:



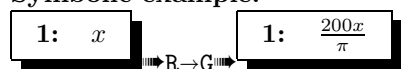
7.4.7 $R \rightarrow G$

Radians to Grads: Converts a real number expressed in radians to its equivalent in grads.

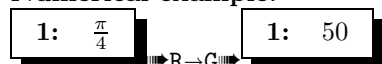
General form:



Symbolic example:



Numerical example:



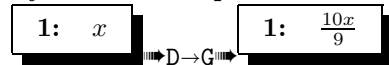
7.4.8 $D \rightarrow G$

Degrees to Grads: Converts a real number expressed in degrees to its equivalent in grads.

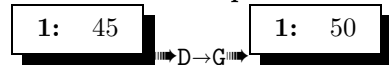
General form:



Symbolic example:



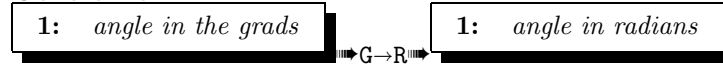
Numerical example:



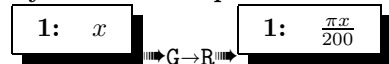
7.4.9 $G \rightarrow R$

Grads to Radians: Converts a real number expressed in grads to its equivalent in radians.

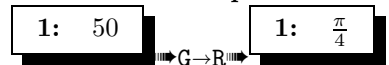
General form:



Symbolic example:



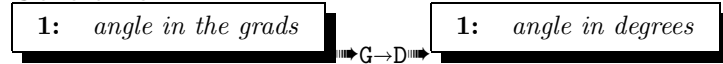
Numerical example:



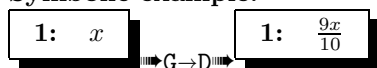
7.4.10 $G \rightarrow D$

Grads to Degrees: Converts a real number expressed in grads to its equivalent in degrees.

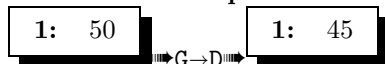
General form:



Symbolic example:



Numerical example:



7.4.11 ANGLE

Complex Number Angle: Returns the angle of a complex number, if a line were to be extended from the complex number to (0,0), formed with the real number line.

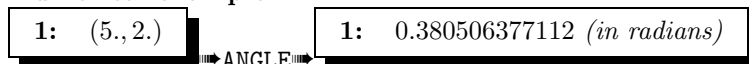
General form:



Symbolic example:



Numerical example:



7.5 HYPER: Hyperbolics

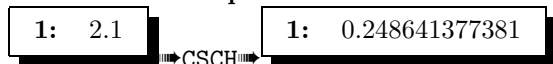
7.5.1 CSCH

Hyperbolic Cosecant: Returns the hyperbolic cosecant of the argument.

General form:



Numerical example:



7.5.2 ACSCH

Inverse Hyperbolic Cosecant: Returns the inverse hyperbolic cosecant of the argument.

General form:



Numerical example:



7.5.3 SECH

Hyperbolic Secant: Returns the hyperbolic secant of the argument.

General form:



Numerical example:



7.5.4 ASECH

Inverse Hyperbolic Secant: Returns the inverse hyperbolic secant of the argument.

General form:



Numerical example:



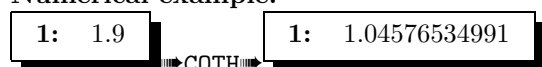
7.5.5 COTH

Hyperbolic Cotangent: Returns the hyperbolic cotangent of the argument.

General form:



Numerical example:



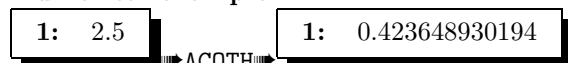
7.5.6 ACOTH

Inverse Hyperbolic Cotangent: Returns the inverse hyperbolic cotangent of the argument.

General form:



Numerical example:



7.6 SYMB: Symbolic Manipulation

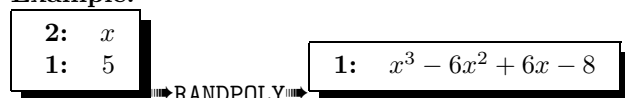
7.6.1 RANDPOLY

Random Polynomial: Given a variable and order, this function generates a random polynomial with integer coefficients in the range $[-9, 9]$.

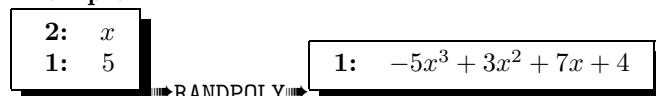
General form:



Example:



Example:



7.6.2 LEFT

This returns the left-hand side of an equation or inequality.

General form:



Symbolic example:



7.6.3 RIGHT

This returns the right-hand side of an equation or inequality.

General form:



Symbolic example:



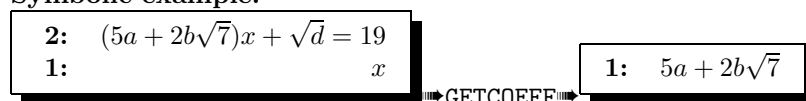
7.6.4 GETCOEFF

Coefficient of a Variable: Returns the coefficient of a variable in an expression. The expression is reduced and simplified before the coefficient is found, so something like $\frac{ax}{b} + \frac{2x}{b} = \frac{z}{b}$ can return a sensible and consistent value as the coefficient of x , in this case $a + 2$.

General form:



Symbolic example:



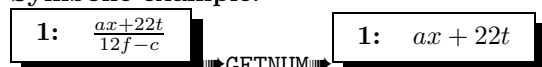
7.6.5 GETNUM

Numerator of a Fraction: Returns the numerator of an expression.

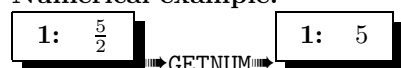
General form:



Symbolic example:



Numerical example:



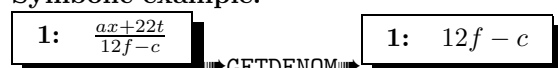
7.6.6 GETDENOM

Denominator of a Fraction: Returns the denominator of an expression.

General form:



Symbolic example:



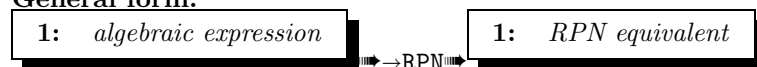
Numerical example:



7.6.7 →RPN

This is an improved version of the →RPN function found in the 48G series' TEACH command, written by Javier Goizueta.

General form:

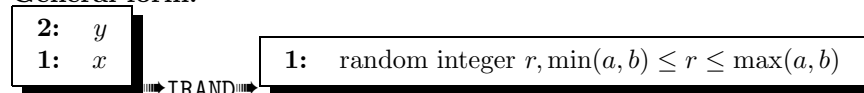


7.7 PROB: Probability

7.7.1 IRAND

Random Integer: Given two numbers a and b , returns a pseudo-random integer between the two, inclusive.

General form:



Example:



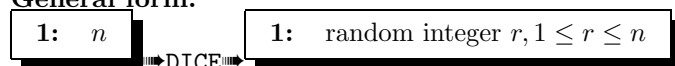
Example:



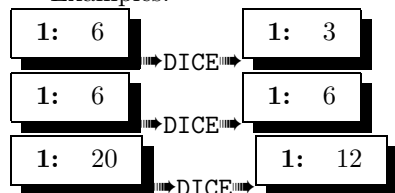
7.7.2 DICE

Dice Simulation: Given a number n , returns a pseudo-random integer between 1 and n . Therefore, $n = 6$ would act like a six-sided dice roll.

General form:



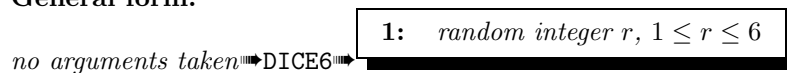
Examples:



7.7.3 DICE6

This function is merely the DICE function, but assuming 6 sides, like most standard dice.

General form:



Example:



7.7.4 COIN

This function randomly returns either 0 or 1, simulating a coin toss.

General form:

no arguments taken \Rightarrow COIN \Rightarrow

1:	<i>randomly 1 (true) or 0 (false)</i>
----	---------------------------------------

Example:

no arguments taken \Rightarrow COIN \Rightarrow

1:	1
----	---

7.7.5 LTPN

Lower-Tail Normal Distribution: Similar to the built-in UTPN, but instead it calculates the lower-tail.

General form:

3:	μ
2:	σ^2
1:	x

 \Rightarrow LTPN \Rightarrow

1:	<i>prob. between $-\infty$ and x</i>
----	--

Numerical example:

3:	3.1
2:	29.9
1:	1.25

 \Rightarrow LTPN \Rightarrow

1:	0.367558543204
----	----------------

7.7.6 RNGPN

Ranged Normal Distribution: Similar to the built-in UTPN, but instead it calculates a range.

General form:

4:	μ
3:	σ^2
2:	y
1:	x

 \Rightarrow RNGPN \Rightarrow

1:	<i>prob. between y and x</i>
----	--

Numerical example:

4:	1.55
3:	90.7
2:	-10.5
1:	3.5

 \Rightarrow RNGPN \Rightarrow

1:	0.478230395145
----	----------------

7.7.7 BIDIST

Binomial Probability Distribution of a Random Variable: This function calculates the probability of success $\Pr(x)$ for n Bernoulli trials, each with a probability of success p , where x denotes the total number of successes in the n trials.

$$\Pr(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

You should note that $0 \leq p \leq 1$, and both x and n should be positive integers.

General form:



Example:



7.7.8 POIDIST

Poisson Probability Distribution of a Random Variable: Returns the probability of a random variable x having a Poisson distribution, which can be found by:

$$\Pr(x) = \frac{e^{-\lambda} \lambda^x}{x!}, x = 0, 1, 2, \dots$$

This is traditionally viewed as a discrete distribution. This distribution was named after the notable French mathematician and physicist Simeon D. Poisson (A.D. 1781–1840).

General form:



Numerical example:



7.8 STAT: Statistics:

The following sample ΣDAT in the following examples. This is inflation data for the United States.

1966	2.92	1975	8.98	1984	4.37	1993	2.98
1967	2.84	1976	5.75	1985	3.54	1994	2.60
1968	4.26	1977	6.62	1986	1.86	1995	2.76
1969	5.29	1978	7.59	1987	3.66	1996	2.96
1970	5.94	1979	11.28	1988	4.12	1997	2.35
1971	4.31	1980	13.48	1989	4.81	1998	1.51
1972	3.31	1981	10.36	1990	5.39	1999	2.21
1973	6.20	1982	6.16	1991	4.22	2000	3.38
1974	11.11	1983	3.21	1992	3.01	2001	2.86

7.8.1 $\Sigma\text{MIDRANGE}$

This function returns the midrange of the statistical data stored in the variable ΣDAT . The midrange of a set of data is defined as:

$$\text{midrange} = \frac{\text{minimum} + \text{maximum}}{2}$$

That is, exactly halfway between the minimum and the maximum.

General form:

no arguments taken $\Rightarrow \Sigma\text{MIDRANGE} \Rightarrow$ **1:** *midrange of the data*

Example:

(assuming the inflation data above is stored in ΣDAT)

no arguments taken $\Rightarrow \Sigma\text{MIDRANGE} \Rightarrow$ **1:** [1983.5, 7.495]

7.8.2 $\Sigma\%\text{TILE}$

Percentile of ΣDAT : Returns the specified percentile of the data in ΣDAT .

General form:

1: *n* $\Rightarrow \Sigma\%\text{TILE} \Rightarrow$ **1:** *nth percentile*

Example:

(assuming the inflation data above is stored in ΣDAT)

1: 25 $\Rightarrow \Sigma\%\text{TILE} \Rightarrow$ **1:** [1974.5, 2.94]

7.8.3 Σ MEDIAN

This function returns the median of the statistical data stored in the variable Σ DAT.

General form:

no arguments taken $\Rightarrow \Sigma$ MEDIAN \Rightarrow

1: median of the data

Example:

(assuming the inflation data above is stored in Σ DAT)

no arguments taken $\Rightarrow \Sigma$ MEDIAN \Rightarrow

1: [1983.5, 4.17]

7.8.4 Σ RANGE

This function returns the range of the statistical data stored in the variable Σ DAT.

General form:

no arguments taken $\Rightarrow \Sigma$ RANGE \Rightarrow

1: range of the data

Example:

(assuming the inflation data above is stored in Σ DAT)

no arguments taken $\Rightarrow \Sigma$ RANGE \Rightarrow

1: [35, 11.97]

7.8.5 ZSCORE

Sample Z-Score: Returns the sample z-score (or sample *standard score*) for a specified value. The z-score is the number of sample standard deviations the specified value is from the mean, relative to the statistical data in Σ DAT.

General form:

1: value

 \Rightarrow ZSCORE \Rightarrow

1: sample z-score

Example:

(assuming the inflation data above is stored in Σ DAT)

1: [1990, 5.39]

 \Rightarrow ZSCORE \Rightarrow

1: [1.62086980814, 6.60723682884]

7.8.6 PZSCORE

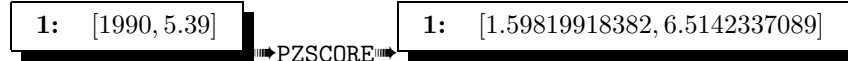
Population Z-Score: Returns the population z-score (or population *standard score*) for a specified value. The z-score is the number of population standard deviations the specified value is from the mean, relative to the statistical data in ΣDAT .

General form:



Example:

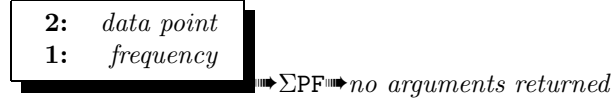
(assuming the inflation data above is stored in ΣDAT)



7.8.7 ΣPF

Frequency $\Sigma+$: Applies the $\Sigma+$ command repeatedly, in effect adding a frequency occurrence of the data point.

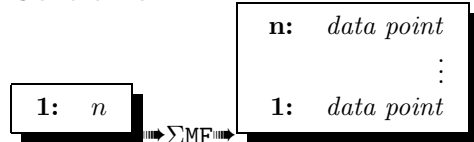
General form:



7.8.8 ΣMF

Frequency $\Sigma-$: Applies the $\Sigma-$ command repeatedly, n times.

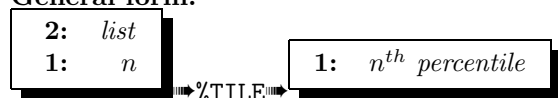
General form:



7.8.9 %TILE

Percentile of a List: Sorts a list, and then returns the value of a specified percentile of the list. For example, typing { some list } 50 %TILE returns the median (50th percentile) of the list. This was an example program in the *HP 48G AUR* [6].

General form:



7.8.10 MEDIAN

Median of a List: Sorts a list, and then returns the median value of the list.

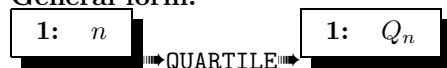
General form:



7.8.11 QUARTILE

Returns the specified quartile of the data in ΣDAT .

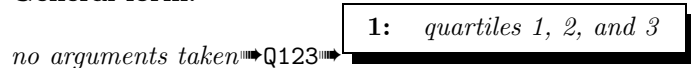
General form:



7.8.12 Q123

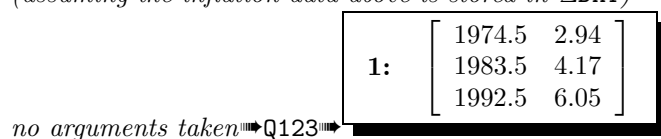
Quartiles 1, 2, and 3: Returns the first, second, and third quartiles of the data in ΣDAT .

General form:



Example:

(assuming the inflation data above is stored in ΣDAT)



7.8.13 IQR

Inter-Quartile Range: Returns the inter-quartile range of the data in ΣDAT . The inter-quartile range is defined as the difference between the first and the third quartiles:

$$IQR = Q_3 - Q_1$$

General form:

no arguments taken \Rightarrow IQR \Rightarrow **1:** *inter-quartile range*

Example:

(assuming the inflation data above is stored in Σ DAT)

no arguments taken \Rightarrow IQR \Rightarrow **1:** [18, 3.11]

7.9 CALC: Calculus and Differential Equations

7.9.1 MDER

Multiple Derivative: Evaluates derivatives of order higher than 1, by successive application of the built-in ∂ function. Therefore, to evaluate $\frac{d^7}{dx^7}(x^{23})$, one does not need to type in $\frac{\partial}{\partial x}(\frac{\partial}{\partial x}(\frac{\partial}{\partial x}(\frac{\partial}{\partial x}(\frac{\partial}{\partial x}(\frac{\partial}{\partial x}(\frac{\partial}{\partial x}(x^{23})))))$). This can't handle fractional or negative derivatives.

General form:

3: $f(x)$
2: x
1: n

\Rightarrow MDIR \Rightarrow **1:** $\frac{d^n}{dx^n}(f(x))$

Symbolic example:

3: x^{12}
2: x
1: 5

\Rightarrow MDIR \Rightarrow **1:** $95040x^7$

7.9.2 EUAPPROX

Euler's Approximation Method: Approximates the values of a function $y(x)$ given y' , and the previous x and y values. This is also known as *the method of tangent lines*. given a step-size h , Euler's method is:

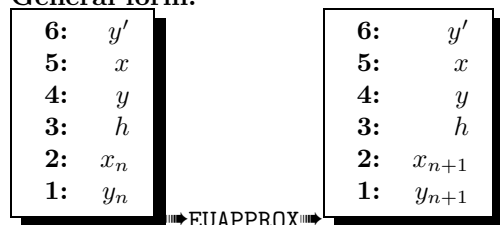
$$x_n = n \cdot h + x_0$$

and

$$y_n = h \cdot y'(x_{n-1}, y_{n-1}) + y_{n-1}$$

The program's output is the input for the next step, so you can just input the initial conditions and repetitively evaluate the function just by pressing the button over and over again.

General form:



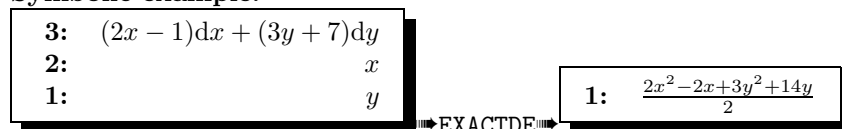
7.9.3 EXACTDE

Exact Differential Equation Solver: Given an exact differential equation, finds the solution equation. An exact differential equation is one in the form $M(x,y)dx + N(x,y)dy = 0$, where $\frac{\partial M(x,y)}{\partial y} = \frac{\partial N(x,y)}{\partial x}$.

General form:



Symbolic example:



7.9.4 WRONSKIAN

Wronskian Determinant: Finds the Wronskian determinant of a set of functions with respect to a variable. The Wronskian is defined as:

$$W(f_1, f_2, \dots, f_n) = \begin{vmatrix} f_1 & f_2 & \dots & f_n \\ f_1' & f_2' & \dots & f_n' \\ \vdots & \vdots & \ddots & \vdots \\ f_1^{(n-1)} & f_2^{(n-1)} & \dots & f_n^{(n-1)} \end{vmatrix}$$

You must supply the list of functions and the variable of differentiation.

General form:



Symbolic example:

2:	$\{e^{3x}, e^{-3x}\}$		
1:	x		
⇒ WRONSKIAN ⇒			
		1:	-6

7.10 VECTR: Vectors

7.10.1 PROJ

Vector Orthogonal Projection: Given a non-zero vector \vec{y} and another vector \vec{x} , computes the orthogonal projection of \vec{y} onto \vec{x} , defined as:

$$\text{proj}_{\vec{y}}\vec{x} = \left(\frac{\vec{x} \cdot \vec{y}}{\vec{y} \cdot \vec{y}} \right) \vec{y}$$

General form:

2:	\vec{y}		
1:	\vec{x}		
⇒ PROJ ⇒			
		1:	$\text{proj}_{\vec{y}}\vec{x}$

Numerical example:

2:	$[7, 6, 2]$		
1:	$[2, 8, 12]$		
⇒ PROJ ⇒			
		1:	$\left[\frac{602}{89}, \frac{516}{89}, \frac{172}{89} \right]$

Numerical example:

2:	$[7., 6., 2.]$		
1:	$[2., 8., 12.]$		
⇒ PROJ ⇒			
		1:	$[6.76404494382, 5.79775280899, 1.93258426966]$

Symbolic example:

2:	$[a, b, c]$		
1:	$[d, e, f]$		
⇒ PROJ ⇒			
		1:	$\left[\frac{a(cf+be+ad)}{a^2+b^2+c^2}, \frac{b(cf+be+ad)}{a^2+b^2+c^2}, \frac{c(cf+be+ad)}{a^2+b^2+c^2} \right]$

7.10.2 ACROSS

Left-Side Inverse Cross Product: For the expression $\vec{a} \times \vec{b} = \vec{c}$, given \vec{b} and \vec{c} , returns a solution to \vec{a} . You should note that this solution is not unique, for there are an infinite amount of solutions.

General form:

2:	\vec{b}		
1:	\vec{c}		
⇒ ACROSS ⇒			
		1:	\vec{a}

7.10.3 BCROSS

Right-Side Inverse Cross Product: For the expression $\vec{a} \times \vec{b} = \vec{c}$, given \vec{a} and \vec{c} , returns a solution to \vec{b} . You should note that this solution is not unique, for there are an infinite amount of solutions.

General form:



7.10.4 UNITV

Unit Vector: Given a vector \vec{v} , calculates a vector in the direction of \vec{v} with a length of 1. This can be found by $\hat{v} = \frac{\vec{v}}{|\vec{v}|}$.

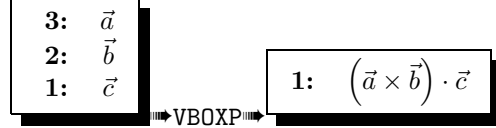
General form:



7.10.5 VBOXP

Vector Box Product: Calculates the vector box product, also known as the triple scalar product, of three vectors.

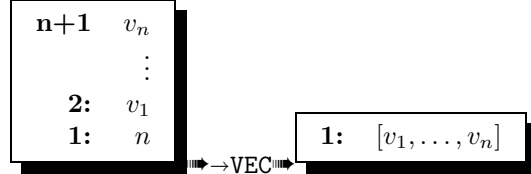
General form:



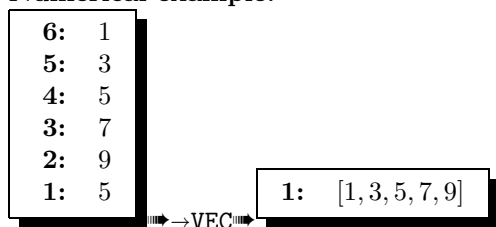
7.10.6 →VEC

Construct a Vector: Given a list of values and the vector length, this will construct the specified vector.

General form:



Numerical example:



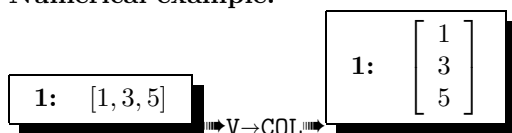
7.10.7 V→COL

Vector to Column Matrix: Converts a vector into a column matrix.

General form:



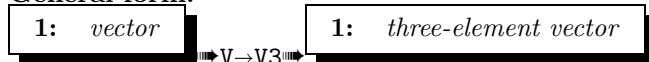
Numerical example:



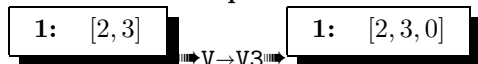
7.10.8 V→V3

Vector to Three-Element Vector: Converts a 1, 2, or 3 element vector into a 3-element vector.

General form:



Numerical example:

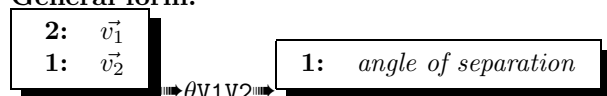


7.10.9 θV1V2

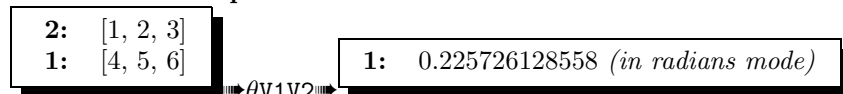
Returns the angle of separation of two vectors. This can generally be found by:

$$\theta = \arccos \left(\frac{\hat{v}_1 \cdot \hat{v}_2}{|\hat{v}_1||\hat{v}_2|} \right)$$

General form:



Numerical example:



7.11 MATRX: Matrices

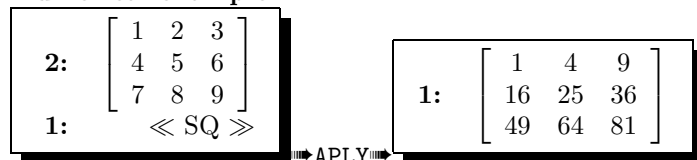
7.11.1 APLY

Apply to an Array: Applies a program to each element of an array. The program applied must take exactly one input and return exactly one output. If output is symbolic result is returned as a “symbolic array” (that is, a list of “row” lists instead of an array of row vectors). This is taken from the HP 48G’s TEACH library.

General form:



Numerical example:



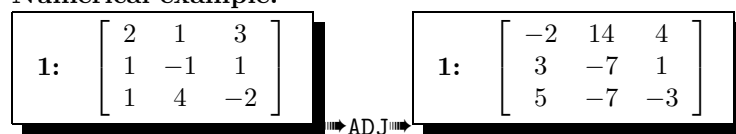
7.11.2 ADJ

Matrix Adjugate (Classical Adjoint): Given a matrix A , returns $\text{adj}A$, the adjugate (or classical adjoint) of A .

General form:



Numerical example:



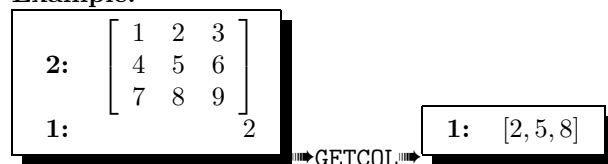
7.11.3 GETCOL

Given a matrix M and a column number c , returns the specified column from M as a vector.

General form:



Example:



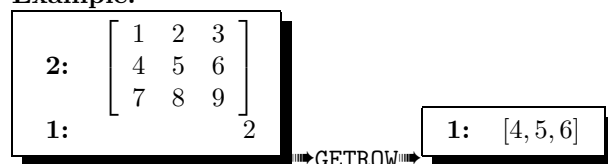
7.11.4 GETROW

Given a matrix M and a row number r , returns the specified row from M as a vector.

General form:

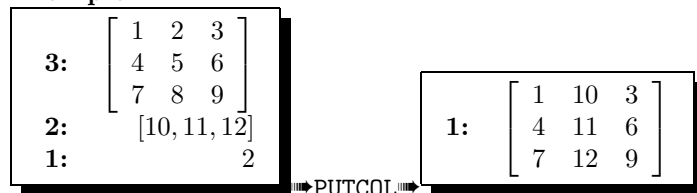


Example:

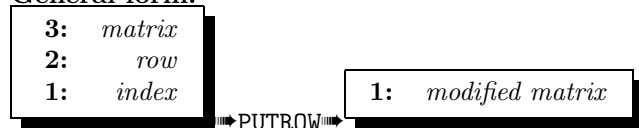
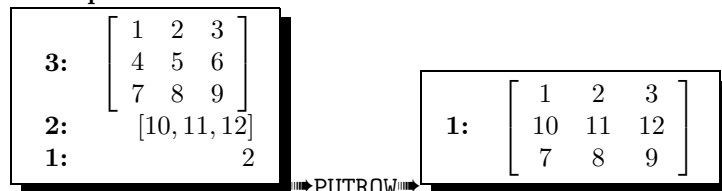


7.11.5 PUTCOL

Given a matrix, a column, and an index, replaces the specified column of the matrix. This is similar to the built-in command COL+, but overwrites the specified column instead of just adding another.

General form:**Example:****7.11.6 PUTROW**

Given a matrix, a row, and an index, replaces the specified row of the matrix. This is similar to the built-in command `ROW+`, but overwrites the specified row instead of just adding another.

General form:**Example:****7.12 LIST: Lists****7.12.1 APPEND**

Append to a List: Given a list l and any other object (even another list) x , appends x to the end of l .

General form:

Example:



7.12.2 PREPEND

Prepend to a List: Given any object (even another list) x and a list l , prepends x to the beginning of l .

General form:



Example:



7.12.3 FLATLIST

Nested List Flattening: Flattens a nested list of lists into a single list which is only one level deep.

General form:



General form:



7.12.4 LoL→L

This command works in a similar manner to FLATLIST, but only applies to one level of a list, whereas FLATLIST applies itself recursively. Unless you specifically want this behavior, you most likely actually want to use FLATLIST instead.

Example:



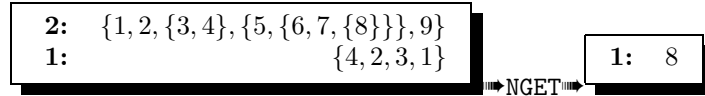
7.12.5 NGET

Nested List GET: Returns the specified element from a nested list.

General form:



Example:



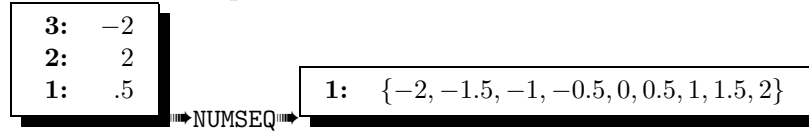
7.12.6 NUMSEQ

Number Sequence: Given a starting value, an ending value, and a step size, this function generates the desired sequence of numbers as a list.

General form:



Numerical example:



7.12.7 ΣSTEP

List Step-wise Summation: Given a list of numbers, outputs a list containing the step-wise summations for each element. Thus, for each element of the output list s relates to the input list l by

$$s_i = \sum_{j=1}^i l_j \forall i \in \{1, 2, \dots, |l| - 1, |l|\}$$

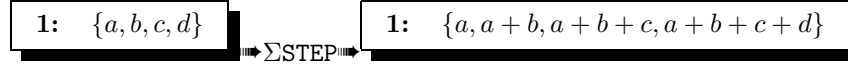
General form:



Numerical example:



Symbolic example:



7.12.8 IISTEP

List Step-wise Product: Given a list of numbers, outputs a list containing the step-wise products for each element. Thus, for each element of the output list s relates to the input list l by

$$s_i = \prod_{j=1}^i l_j \forall i \in \{1, 2, \dots, |l| - 1, |l|\}$$

General form:



Numerical example:



Symbolic example:

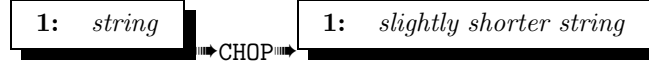


7.13 CHARS: Characters and Strings

7.13.1 CHOP

Chop a String: This function removes the last character of a string.

General form:



Example:



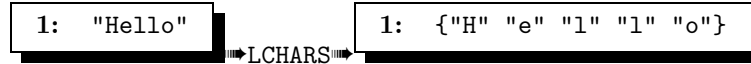
7.13.2 LCHARS

String to List of Characters: Given a string as input, this function returns a list containing each character as a separate item. In order to reverse this, simply use the built-in Σ LIST function.

General form:



Example:



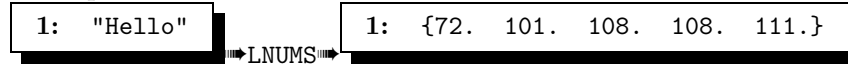
7.13.3 LNUMS

String to List of Numbers: Given a string as input, this function returns a list containing each character's numerical value as a separate item. Use NSTR to reverse this.

General form:



Example:



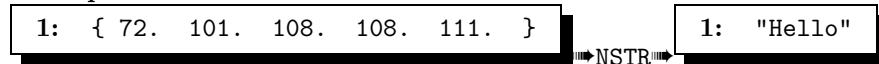
7.13.4 NSTR

List of Numbers to String: This function is the inverse for the LNUMS function. Given a list of numerical values for characters, this function returns the appropriate string.

General form:



Example:



7.13.5 CAESAR

Caesar Shift: Given a string and a shift value, shifts the characters in the string by the specified shift, in \mathbb{Z}_{256} . Julius Caesar supposedly used such a method on occasion, with a shift of 3 (-3 to decrypt). Take heed that this is by no means a secure method of encryption, but rather just an historical curiosity.

General form:



Example:



7.13.6 VIGENERE

Vigenère Encryption: Given a plaintext string and a passphrase, returns the appropriate ciphertext string, in \mathbb{Z}_{256} . Take heed that this is by no means a secure method of encryption, but rather just an historical curiosity.

General form:



Example:



7.13.7 DEVIGENERE

Vigenère Decryption: Given a ciphertext string and a passphrase, returns the appropriate plaintext string, in \mathbb{Z}_{256} . Take heed that this is by no means a secure method of encryption, but rather just an historical curiosity.

General form:



Example:

2: "µäüë××ÑÛÜæ∈éÔÓ■ÛÜ"	
1: "password"	1: "Trithemius really"

→VIGENERE→

7.14 PROG: Programming

7.14.1 NMENU

Nesting menu function. This program works almost identically to the built-in TMENU function, except that it allows for sub-menus. A sub-menu entry is a list containing three items:

1. A string containing the title to be shown for the sub-menu.
2. A string that is "SUBMENU".
3. A CST-like menu list.

You can have sub-menus within sub-menus as deeply as you wish.

Example:

1: {SIN COS TAN {"HYPER" "SUBMENU" {SINH COSH TANH}}}

→NMENU→*soft-menu modified*

See NCST1278 for a more complete example. Also, this will run much faster if you save the nested custom menu somewhere, and just pass it the unresolved variable name.

Example:

(assuming NCST holds a nested custom menu.)

1: 'NCST'

→NMENU→*soft-menu modified*

7.14.2 FOLDERGROB

Given a string, returns a GROB of folder for use in menus.

7.14.3 NCST1278

This is the CST-like list used by the program Menu1278. It is here as an example of how to use the NMENU function.

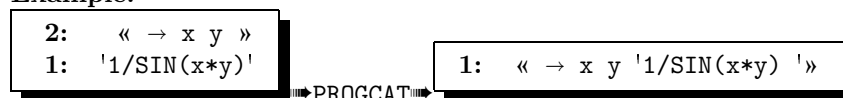
7.14.4 PROGCAT

Program Concatenation: Given two programs, concatenates them into a single program, or will build a program out of two non-program items.

Example:



Example:



Example:



Example:



7.14.5 NULLDIR

Changes to the hidden, “null” directory.

7.15 TIME: Date and Time

7.15.1 DAY

Returns the current day of the month as an integer.

7.15.2 MONTH

Returns the current month as an integer.

7.15.3 YEAR

Returns the current year as an integer.

7.15.4 WEEKDAY

Returns the current day of the week, as the standard 3-letter abbreviation.

7.16 MISC: Miscellaneous

7.16.1 FWGHT

Chemical Formula Weight Program: This program calculates the formula weight of a chemical formula. Since there are no subscripts or superscripts in the HP 49G+, you must enter the formula in an algebraic form. For example ethanol, $\text{CH}_3\text{CH}_2\text{OH}$, would be entered as $C + H3 + C + H2 + O + H$. Note that, just like everything else in the HP 49G+, this is case sensitive. Therefore O is oxygen but o is not, and sodium is Na but not na, NA or nA.

General form:

1: <i>chemical formula</i>	→FWGHT→	1: <i>formula weight</i>
----------------------------	---------	--------------------------

Example:

1: $C + H3 + C + H2 + O + H$	→FWGHT→	1: $46.06904 \frac{g}{mol}$
------------------------------	---------	-----------------------------

7.16.2 NPV

Net Present Value: Given a list of cash flows, and a preferred rate of return, calculates the net present value.

General form:

2: <i>list of cash flows</i>	→NPV→	1: <i>net present value</i>
1: <i>preferred rate of return</i>		

Numerical example:

This is taken from the *HP-12C Owner's Handbook*, page 68. An investor has an opportunity to buy a duplex for \$80,000 and would like a return of at least 13%. He expects to lose \$500 in the first year, make \$4,500 in the second year, \$5,500 in the third, \$4,500 in the fourth, and then sell the property for \$130,000 after five years.

2: $\{-80000, -500, 4500, 5500, 4500, 130000\}$	→NPV→	1: 212.184061
1: 13		

Since the NPV is positive, the investment will return greater than the desired 13%. A negative NPV indicates a return lower than desired, and the more negative or positive, then the greater a loss or gain relative to that which is desired. The *internal rate of return* (IRR) is the rate where the NPV is 0, and is the actual return of the investment. The IRR for this example is 13.06289%.

7.16.3 STOSTK

Store the Entire Stack: Stores the entire stack in the variable specified in level 1.

7.16.4 RCLSTK

Recall a Stored Stack: Recalls the stack stored in the variable specified by level 1.

Chapter 8

XLIB Numbers

XLIB 1278 000: Menu1278	XLIB 1278 030: DICE6
XLIB 1278 001: NCST1278	XLIB 1278 031: D→G
XLIB 1278 002: %TILE	XLIB 1278 032: D→MODE
XLIB 1278 003: ACOT	XLIB 1278 033: ERF
XLIB 1278 004: ACOTH	XLIB 1278 034: ERFC
XLIB 1278 005: ACROSS	XLIB 1278 035: EUAPPROX
XLIB 1278 006: ACSC	XLIB 1278 036: EXACTDE
XLIB 1278 007: ACSCH	XLIB 1278 037: FIBON
XLIB 1278 008: ADJ	XLIB 1278 038: FLATLIST
<i>XLIB 1278 009: (AFFINE)</i>	XLIB 1278 039: FOLDERGROB
XLIB 1278 010: ANGLE	XLIB 1278 040: FRESC
XLIB 1278 011: APLY	XLIB 1278 041: FRESS
XLIB 1278 012: ARCLEN	XLIB 1278 042: FWGHT
XLIB 1278 013: ASEC	XLIB 1278 043: GETCOEFF
XLIB 1278 014: ASECH	XLIB 1278 044: GETCOL
XLIB 1278 015: AVGRC	XLIB 1278 045: GETDENOM
XLIB 1278 016: BCROSS	XLIB 1278 046: GETNUM
XLIB 1278 017: BETA	XLIB 1278 047: GETROW
XLIB 1278 018: CAESAR	XLIB 1278 048: G→D
XLIB 1278 019: PREPEND	XLIB 1278 049: G→MODE
XLIB 1278 020: CHOP	XLIB 1278 050: G→R
XLIB 1278 021: COIN	XLIB 1278 051: HAV
XLIB 1278 022: COT	XLIB 1278 052: IQR
XLIB 1278 023: COTH	XLIB 1278 053: IRAND
XLIB 1278 024: COVERS	<i>XLIB 1278 054: (IRR)</i>
XLIB 1278 025: CSC	XLIB 1278 055: LCHARS
XLIB 1278 026: CSCH	XLIB 1278 056: LD
XLIB 1278 027: DAY	XLIB 1278 057: LEFT
XLIB 1278 028: DEVIGENERE	XLIB 1278 058: LNUMS
XLIB 1278 029: DICE	XLIB 1278 059: LOGYX

XLIB 1278 060: LoL→L	XLIB 1278 091: UNITV
XLIB 1278 061: MDER	XLIB 1278 092: VBOXP
XLIB 1278 062: MEDIAN	XLIB 1278 093: VERS
XLIB 1278 063: MONTH	XLIB 1278 094: VIGENERE
XLIB 1278 064: RNGPN	XLIB 1278 095: NULLDIR
XLIB 1278 065: NGET	XLIB 1278 096: V→COL
XLIB 1278 066: NMENU	XLIB 1278 097: V→V3
XLIB 1278 067: NPV	XLIB 1278 098: WRONSKIAN
XLIB 1278 068: NSTR	XLIB 1278 099: YEAR
XLIB 1278 069: NUMSEQ	XLIB 1278 100: ZRELPRIME
XLIB 1278 070: ORDMULT	XLIB 1278 101: ZSCORE
XLIB 1278 071: PFINV	XLIB 1278 102: ZSEQ
XLIB 1278 072: PFUNC	XLIB 1278 103: PUTCOL
XLIB 1278 073: LTPN	XLIB 1278 104: ΣMEDIAN
XLIB 1278 074: BIDIST	XLIB 1278 105: ΣMF
XLIB 1278 075: PROGCAT	XLIB 1278 106: ΣMIDRANGE
XLIB 1278 076: PROJ	XLIB 1278 107: ΣPF
XLIB 1278 077: POIDIST	XLIB 1278 108: ΣRANGE
XLIB 1278 078: PZSCORE	XLIB 1278 109: ΣSTEP
XLIB 1278 079: Q123	XLIB 1278 110: →D
XLIB 1278 080: QUARTILE	XLIB 1278 111: →G
XLIB 1278 081: RANDPOLY	XLIB 1278 112: →R
XLIB 1278 082: RELPRIME	XLIB 1278 113: →RPN
XLIB 1278 083: RIGHT	XLIB 1278 114: θV1V2
XLIB 1278 084: RSDEV	XLIB 1278 115: APPEND
XLIB 1278 085: R→G	XLIB 1278 116: STOSTK
XLIB 1278 086: R→MODE	XLIB 1278 117: RCLSTK
XLIB 1278 087: SEC	XLIB 1278 118: →VEC
XLIB 1278 088: SECH	XLIB 1278 119: Σ%TILE
XLIB 1278 089: WEEKDAY	XLIB 1278 120: PUTROW
XLIB 1278 090: UNITSTEP	

Bibliography

- [1] Hewlett-Packard.
HP 49G+ Graphing Calculator User's Guide.
- [2] Hewlett-Packard.
HP 49G+ Graphing Calculator User's Manual.
HP Part No. F2228-90020.
- [3] Hewlett-Packard.
HP 49G User's Guide.
HP Part No. F1633-90001.
- [4] Hewlett-Packard.
HP 49G Advanced User's Guide.
HP Part No. F1633-90401.
- [5] Hewlett-Packard.
HP 48G Series User's Guide.
Edition 8, December 1994.
HP Part No. 00048-90126.
- [6] Hewlett-Packard.
HP 48G Series Advanced User's Reference Manual.
Edition 4, December 1994.
HP Part No. 00048-90136.

Index

- IISTEP, 47
- II
 - step-wise, 47
- ΣMEDIAN, 34
- ΣMF, 35
- ΣMIDRANGE, 33
- ΣPF, 35
- ΣRANGE, 34
- ΣSTEP, 46
- Σ%TILE, 33
- Σ
 - step-wise, 46
- Σ+, 35
- Σ−, 35
- D, 23
- G, 23
- R, 23
- RPN, 29
- VEC, 40
- θV1V2, 41
- ×, 39, 40
- %TILE, 35

- ACOT, 21
- ACOTH, 27
- ACROSS, 39
- ACSC, 20
- ACSCH, 25
- ADJ, 42
- ANGLE, 25
- angle, 41
- angles, 22–25
- APLY, 42
- APPEND, 44
- append
 - list, 44
- approximation
 - Euler’s, 37
- arc cosecant, 20
- arc cotangent, 21
- arc hyperbolic cosecant, 25
- arc hyperbolic cotangent, 27
- arc hyperbolic secant, 26
- arc length, 15
- arc secant, 20
- ARCLEN, 15
- ASEC, 20
- ASECH, 26
- average rate of change, 16
- AVGRC, 16

- BCROSS, 40
- BETA, 16
- BIDIST, 32
- binomial distribution, 32

- CAESAR, 49
- Caesar shift, 49
- chemistry
 - formula weight, 52
 - pH, 18
- CHOP, 47
- COIN, 30
- complementary error function, 17
- complex numbers, 25
- concatenation
 - programs, 50
- cosecant, 19
 - arc (inverse), 20
 - hyperbolic, 25
 - arc (inverse), 25
- COT, 20
- cotangent, 20
 - arc (inverse), 21

- hyperbolic, 26
 - arc (inverse), 27
- COTH, 26
- COVERS, 21
- coversine, 21
- CSC, 19
- CSCH, 25
- D→G, 24
- D→MODE, 22
- DAY, 51
- day, 51
 - weekday, 51
- degrees, 22–24
- denominator, 29
- derivative, 37
- DEVIGENERE, 49
- DICE, 30
- DICE6, 30
- differential equations
 - exact, 38
- directory
 - GROB, 50
 - hidden “null” directory., 51
- encryption
 - Caesar, 49
 - Vigenère, 49
- ERF, 17
- ERFC, 17
- EUAPPROX, 37
- Euler’s approximation, 37
- EXACTDE, 38
- FIBON, 13
- Fibonacci numbers, 13
- financial
 - net present value, 52
- FLATLIST, 45
- FOLDERGROB, 50
- formula weight, 52
- frequency statistics, 35
- FRESC, 19
- fresnel C integral, 19
- fresnel S integral, 19
- FRESS, 19
- FWGHT, 52
- G→D, 24
- G→MODE, 22
- G→R, 24
- Gaussian error function, 17
 - complimentary, 17
- GETCOEFF, 28
- GETCOL, 43
- GETDENOM, 29
- GETNUM, 28
- GETROW, 43
- grads, 22–24
- GROB
 - folders (directories), 50
- HAV, 22
- haversine, 22
- hyperbolic cosecant, 25
 - arc (inverse), 25
- hyperbolic cotangent, 26
 - arc (inverse), 27
- hyperbolic secant, 26
 - arc (inverse), 26
- inflation data, 33
- integer
 - random, 29
- IQR, 36
- IRAND, 29
- LCHARS, 48
- LD, 18
- LEFT, 27
- list, 48
 - append, 44
 - element retrieval, 46
 - flattened, 45
 - number sequence, 46
 - prepend, 45
 - step-wise product, 47
 - step-wise summation, 46
- LNUMS, 48
- logarithm, 17
 - base 2, 18
- logarithmus dualis, 18

LOGYX, 17
 LoL→L, 45
 LTPN, 31
 matrix
 adjucate (classical adjoint), 42
 column, 41, 43
 row, 43, 44
 MDER, 37
 MEDIAN, 36
 median, 34, 36
 MENU, 50
 midrange, 33
 MONTH, 51
 month, 51

 NCST1278, 50
 net present value, 52
 NGET, 46
 NMENU, 50
 NMENU, 50
 normal distribution, 31
 NPV, 52
 NSTR, 48
 NULLDIR, 51
 numerator, 28
 NUMSEQ, 46

 ORDMULT, 15

 percentile, 33, 35
 PFINV, 18
 PFUNC, 18
 pH, 18
 POIDIST, 32
 Poisson distribution, 32
 polynomial
 random, 27
 PREPEND, 45
 prepend
 list, 45
 product
 step-wise, 47
 PROGCAT, 50
 program
 concatenation, 50

 PROJ, 39
 projection, 39
 PUTCOL, 43
 PUTROW, 44
 PZSCORE, 35

 Q123, 36
 QUARTILE, 36
 quartile, 36

 R→G, 23
 R→MODE, 22
 radians, 22–24
 random, 30
 integer, 29
 polynomial, 27
 RANDPOLY, 27
 range, 34
 RCLSTK, 53
 relative primality, 14
 RELPRIME, 14
 RIGHT, 28
 RNGPN, 31
 RPN, 29

 SEC, 20
 secant, 20
 arc (inverse), 20
 hyperbolic, 26
 arc (inverse), 26
 SECH, 26
 sequence
 list of numbers, 46
 stack, 53
 statistical distribution
 binomial, 32
 normal, 31
 Poisson, 32
 STOSTK, 53
 string, 47, 48
 summation
 step-wise, 46

 time
 day, 51
 month, 51

- weekday, 51
 - year, 51
- TMENU, 50
- UNITSTEP, 13
- UNITV, 40
- V→COL, 41
- V→V3, 41
- VBOXP, 40
- vector, 39–41
 - angle of separation, 41
 - box product, 40
 - projection, 39
 - unit, 40
- VERS, 21
- versine, 21
- Vigenère encryption, 49
- VIGENERE, 49
- WEEKDAY, 51
- weekday, 51
- WRONSKIAN, 38
- Wronskian determinant, 38
- XLIB, 54
- YEAR, 51
- year, 51
- z-score
 - population, 35
 - sample, 34
- \mathbb{Z}_n , 14
- \mathbb{Z}_n^* , 14
- ZRELPRIME, 14
- ZSCORE, 34
- ZSEQ, 14