HP 49G Advanced Users Guide

Volume 1

Part A:

Computer Algebra Commands

Introduction

This volume details the computer algebra operations that are available on the HP 49G.

For each operation, the following details are provided:

Type: Function or command. Functions can be used as a part of an algebraic objects and commands cannot. When working with functions or commands within Equation Writer:

- When you apply a function to an expression, the function appears as part of the expression. You need to ensure that the expression is selected, then press ↱ EVAL to apply the function to the selection.
- When you apply a command to an expression in Equation Writer, it is evaluated immediately.

Description: A description of the operation.

Access: The menu or choose-list on which an operation can be found, and the keys that you press to access it. If the operation is on a sub-menu, the sub-menu name is in SMALL CAPITALS after the keys.

Input: The input argument or arguments that the operation needs. If the operation uses more than one input argument, details of the arguments and the order in which you supply them are provided: argument order for algebraic mode and stack order for RPN mode.

Output: The output that the operation produces.

- In RPN mode, the outputs are placed on the stack.
- In algebraic mode, the outputs are written to a list.

As with the input arguments, the outputs for both algebraic and RPN mode are described.

Flags: Details of how flag settings affect the operation of the function or command.

Example: An example of the function or command.

See also: Related functions or commands.

Computer algebra command categories

Algebra commands

Arithmetic commands

Arithmetic Integer commands

Arithmetic Polynomial commands

Arithmetic Modulo commands

Calculus commands

Derivation and integration commands

Limits and series commands

Differential equations commands

Exp and Lin commands

Matrix-related commands

Create

Operations

Quadratic form

Linear systems

Eigenvector

Symbolic solve commands

Trigonometry commands

Alphabetical command list

The following pages contain the commands in alphabetical order. See "Computer algebra command categories" on page 3 to view the commands in the order that they appear on the menus.

ABCUV

Type: Command

Description: Returns a solution in polynomials u and v of $au+bv=c$ where a and b are polynomials, and c is a value.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 3/Argument 1: The polynomial corresponding to a.
Level 2/Argument 2: The polynomial corresponding to b.
Level 1/Argument 3: The value corresponding to c.

Output: Level 2/Item 1: The solution corresponding to u.
Level 1/Item 2: The solution corresponding to v.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find a solution in polynomials u and v for the following equation:

$(x^2 + x + 1)u + (x^2 + 4)v = 13$

Command:
```
ABCUV(X^2+X+1,X^2+4,13)
```

Result:
```
{-(X+3),X+4}
```

See also: IABCUV
EGCD

ACOS2S

Type: Command

Description: Transforms an expression by replacing acos(x) in subexpressions with $\pi/2$–asin(x).

Access: Trigonometry, ↱ TRIG

Input: The expression to transform.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:

$$arc\cos\left(\frac{2}{3}\right)+arc\cos(x)$$

Command: `ACOS2S(ACOS(2/3)+ACOS(X))`

Result: `π/2-ASIN(2/3)+π/2-ASIN(X)`

See also: ASIN2C
ASIN2T
ATAN2S

ADDTMOD

Type: Function

Description: Adds two expressions or values, modulo the current modulus.

Access: Arithmetic, ↰ (ARITH) MODULO

Input: Level 2/Argument 1: The first expression.
Level 1/Argument 2: The second expression.

Output: The sum of the two expressions, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Express the result of the following addition in modulo 7.
$(x^2+3x+6)+(9x+3)$
Note: Use the CAS modes input form to set the modulo to 7

Command: `ADDTMOD(X^2+3*X+6,9*X+3)`

Result: `X^2-2*X+2`

ASIN2C

Type: Command

Description: Transforms an expression by replacing asin(*x*) subexpressions with π/2–acos(*x*) subexpressions.

Access: Trigonometry, ↱ TRIG

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

ASIN2T

Type: Command

Description: Transforms an expression by replacing asin(*x*) subexpressions with the following:

$$\text{atan}\left(\frac{x}{\sqrt{1-x^2}}\right)$$

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

ATAN2S

Type: Command

Description: Transforms an expression by replacing atan(*x*) subexpressions with the following:

$$\text{asin}\left(\frac{x}{\sqrt{x^2+1}}\right)$$

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –flag –03 clear).

AXL

Type: Command

Description: Converts a list to an array, or an array to a list.

Access: Convert, ↰ CONVERT

Input: A list or an array.

Output: If the input is a list, returns the corresponding array. If the input is an array, returns the corresponding list.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Convert the following matrix to a list:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Command: `AXL([[0,1][1,0]])`

Result: `{{1,0},{0,1}}`

See also: AXM

AXM

Type: Command

Description: Converts a numeric array to a symbolic matrix.

Access: Matrices, ↰ MATRICES OPERATIONS

Input: An array.

Output: The corresponding matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: AXL
AXQ

AXQ

Type: Command

Description: Converts a square matrix into the associated quadratic form.

Access: Convert, ↰ CONVERT

Input: Level 2/Argument 1: An $n \times n$ matrix.
Level 1/Argument 2: A vector containing n variables.

Output: Level 2/Item 1: The corresponding quadratic form.
Level 1/Item 2: The vector containing the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the quadratic form, expressed in terms of x, y, and z, associated with the following matrix:

$$\begin{bmatrix} 3 & 6 & 0 \\ 2 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Command:
```
AXQ([[3,6,0][2,4,1][1,1,1]],[X,Y,Z])
```

Result:
```
{3*X^2+(8*Y+Z)*X+(4*Y^2+2*Z*Y+Z^2),[X,Y,Z]}
```

See also: GAUSS
QXA

CASCFG

Type: Command

Description: Restores the default CAS mode settings. This command is equivalent to pressing RESET when the CAS Modes input form is displayed.

Access: CAT CASCFG

CHINREM

Type: Command

Description: Solves a system of simultaneous polynomial congruences in the ring Z[*x*].

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: A vector of the first congruence (expression and modulus).
Level 1/Argument 2: A vector of the second congruence (expression and modulus).

Output: A vector of the solution congruence (expression and modulus).

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following simultaneous congruences for the polynomial *u*:

$u \equiv x^2 + 1 \pmod{x+2}$

$u \equiv x - 1 \pmod{x+3}$

Command: `CHINREM([X^2+1,X+2],[X-1,X+3])`

Result: `[X^3+2*X^2+5,-(X^2+5*X+6)]`

See also: ICHINREM

CURL

Type: Function

Description: Returns the curl of a three-dimensional vector function.

Access: Calculus, ↰ CALC DERIV AND INTEG

Input: Level 2/Argument 1: A three-dimensional vector function of three variables.
Level 1/Argument 2: An array comprising the three variables.

Output: The curl of the vector function with respect to the specified variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the curl of the following vector function:

$' = x^2y\underset{\sim}{i} + x^2y\underset{\sim}{j} + y^2z\hat{k}$

Command: `CURL([X^2*Y, X^2*Y, Y^2*Z],[X,Y,Z])`

Result: `[2*X*Y,X^2,Y^2]`

See Also: DIV
HESS

DERIV

Type: Function

Description: Returns the partial derivatives of a function, with respect to the specified variables.

Access: Calculus, ↰ CALC DERIV. & INTEG

Input: Level 2/Argument 1: A function or a list of functions.
Level 1/Argument 2: A variable, or a vector of variables.

Output: The derivative, or a vector of the derivatives, of the function or functions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the gradient of the following function of the spatial variables x, y, and z:

$2x^2y + 3y^2z + zx$

Command:
```
DERIV(2*X^2*Y+3*Y^2*Z+Z*X, [X,Y,Z])
EXPAND(ANS(1))
```

Result: `[4*Y*X+Z,2*X^2+6*Z*Y,X+3*Y^2]`

See also: DERVX

DERVX

Type: Function

Description: Returns the derivative of a function with respect to the current variable.

Access: Calculus, ↰ CALC DERIV. & INTEG.

Input: The function or list of functions to be differentiated.

Output: The derivative, or a vector of the derivatives, of the function or functions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: DERIV

DESOLVE

Type: Command

Description: Solves certain first-order ordinary differential equations with respect to the current variable.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: A first-order differential equation.
Level 1/Argument 2: The function to solve for.

Output: The solution to the equation, either y as a function of x or x as a function of y, or x and y as functions of a parameter.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following differential equation:

$$y'(x) + 2y(x) = e^{3x}$$

Command:
```
DESOLVE(d1Y(X)+2*Y(X)=EXP(3*X),Y(X))
```

Result:
```
{Y(X)=(1/5*EXP(5*X)+C0*)(1/EXP(X)^2)}
```

See also: LDEC

DIV

Type: Command

Description: Returns the divergence of a vector function.

Access: Calculus, ↰ CALC DERIV. & INTEG.

Input: Level 2/Argument 1: An array representing a vector function.
Level 1/Argument 2: An array containing the variables.

Output: The divergence of the vector function with respect to the specified variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the divergence of the following vector function:

$$v = x^2y\underset{\sim}{i} + x^2y\underset{\sim}{j} + y^2z\underset{\sim}{k}$$

Command: `DIV([X^2*Y, X^2*Y, Y^2*Z],[X,Y,Z])`

Result: `Y*2*X+X^2+Y^2`

See also: CURL, HESS

DIV2

Type: Command

Description: Performs euclidean division on two expressions. Step-by-step mode is available with this command.

Access: Arithmetic, ↰ (ARITH) POLYNOMIAL

Input: Level 2/Argument 1: The dividend.
Level 1/Argument 2: The divisor.

Output: Level 2/Item 1: The quotient.
Level 1/Item 2: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Perform the following division:

$$\frac{x^2 + x + 1}{2x + 4}$$

Command: `DIV2(X^2+X+1,2*X+4)`

Result: `{1/2(X-1),3}`

DIV2MOD

Type: Command

Description: Performs euclidean division on two expressions modulo the current modulus.

Access: Arithmetic, ↰ (ARITH) MODULO

Input: Level 2/Argument 1: The dividend.
Level 1/Argument 2: The divisor.

Output: Level 2/Item 1: The quotient.
Level 1/Item 2: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the result of $\frac{x^3+4}{x^2-1}$, modulo the default modulus, 3.

Command: `DIV2MOD(X^3+4,X^2-1)`

Result: `{X X+1}`

DIVIS

Type: Command

Description: Returns a list of divisors of a polynomial or an integer.

Access: Arithmetic, ↰ ARITH

Input: A polynomial or an integer.

Output: A list containing the expressions or integers that exactly divide into the input.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the divisors of the following polynomial:

$x^2 + 3x + 2$

Command: `DIVIS(X^2+3*X+2)`

Result: `{1,X+1,X+2,X^2+3*X+2}`

See also: `DIV2`

DIVMOD

Type: Function

Description: Divides two expressions modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The dividend.
Level 1/Argument 2: The divisor.

Output: The quotient of the terms modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Modulo the default modulus, 3, divide $5x^2+4x+2$ by x^2+1.

Command: `DIVMOD(5*X^2+4*X+2,X^2+1)`

Result: `-((X^2-X+1)/X^2+1))`

DIVPC

Type: Command

Description: Returns a Taylor polynomial for the quotient of two expressions.

Access: Calculus, ↰ CALC LIMITS & SERIES

Input: Level 3/Argument 1: The numerator expression.
Level 2/Argument 2: The denominator expression.
Level 1/Argument 3: The degree of the Taylor polynomial.

Output: The Taylor polynomial at $x = 0$ of the quotient of the two expressions, to the specified degree.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the fourth degree Taylor polynomial for the following:

$$\frac{x^3 + 4x + 12}{11x^{11} + 1}$$

Command: `DIVPC(X^3+4*X+12,11*X^11+1,4)`

Result: `12+4*X+X^3`

EGCD

Type: Command

Description: Given two polynomials, a and b, returns polynomials u, v, and c where:
$au+bv=c$
In the equation, c is the greatest common divisor of a and b.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: The expression corresponding to a in the equation.
Level 1/Argument 2: The expression corresponding to b in the equation.

Output: Level 3/Item 1: The result corresponding to c in the equation.
Level 2/Item 2: The result corresponding to u in the equation.
Level 1/Item 3: The result corresponding to v in the equation.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the polynomials for u, v, and c, where c is the greatest common divisor of a and b such that:

$$u(x^2 + 1) + v(x - 1) = c$$

Command: `EGCD(X^2+1,X-1)`

Result: `{2,1,-(X+1)}`

See also: IEGCD
ABCUV

EPSX0

Type: Function

Description: Replaces all coefficients in a polynomial that have an absolute value less than that held in the variable EPS, with 0. The value in EPS must be less than 1.

Access: Catalog, CAT

Input: A polynomial.

Output: The polynomial with conforming coefficients replaced with 0.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

EULER

Type: Function

Description: For a given integer, returns the number of integers less than the integer that are co-prime with the integer. (Euler's Φ function.)

Access: ↰ (ARITH) INTEGER

Input: A non-negative integer.

Output: The number of positive integers, less than, and co-prime with, the integer.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

EXLR

Type: Command

Description: Returns the left- and right-hand sides of an equation as discrete expressions.

Access: Catalog, (CAT)

Input: An equation.

Output: Level 2/Argument 1: The expression to the left of the "=" sign in the original equation.
Level 1/Argument 2: The expression to the right of the "=" sign in the original equation

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Split the following equation into its two component expressions:
$\sin(x)=5x+y$

Command: `EXLR(SIN(X)=5*X+Y)`

Result: `{SIN(X) 5*X+Y}`

EXPAN

Type: Command

Description: Expands and simplifies an algebraic expression. This command is identical to the EXPAND command. It is included to ensure backward-compatibility with the HP 48-series calculators.

Access: Catalog, CAT

Input: An expression

Output: The expanded and simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: EXPAND

EXPAND

Type: Command

Description: Expands and simplifies an algebraic expression.

Access: Algebra, ↱ ALG

Input: An expression, or an array of expressions.

Output: The expanded and simplified expression or array of expressions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:

$$\frac{(x^2 + 2x + 1)}{x + 1}$$

Command: `EXPAN((X^2+2*X+1)/(X+1))`

Result: `X+1`

See also: EXPAN

EXPANDMOD

Type: Function

Description: Expands and simplifies an algebraic expression, modulo the current modulus.

Access: ↰ (ARITH) MODULO

Input: An expression.

Output: The expanded and simplified expression modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Expand the following expression and give the result modulo 3 (the default modulo setting):
$(x + 3)(x + 4)$

Command:
```
EXPANDMOD((X+3)*(X+4))
```

Result:
```
X^2+X
```

EXPLN

Type: Command

Description: Transforms the trigonometric terms in an expression to exponential and logarithmic terms.

Access: Convert, ↰ (CONVERT)

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Complex mode must be set (flag –103 set).

Example: Transform the following expression and simplify the result using the EXPAND command:

$2\cos(x^2)$

Command:
```
EXPLN(2*COS(X^2))
EXPAND(ANS(1))
```

Result: $\frac{\mathrm{EXP}(\dot{i}X^2)^2 + 1}{\mathrm{EXP}(\ddot{i}X^2)}$

See also: SINCOS

FACTOR

Type: Command

Description: Factorizes a polynomial or an integer:

- The function expresses a polynomial as the product of irreducible polynomials.
- The function expresses an integer as the product of prime numbers.

Access: Algebra, ↱ ALG

Input: An expression or an integer.

Output: The factorized expression, or the integer expressed as the product of prime numbers.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Factorize the following:

$x^2 + 5x + 6$

Command: `FACTOR(X^2+5*X+6)`

Result: `(X+2)(X+3)`

See also: EXPAND

FACTORMOD

Type: Function

Description: Factorizes a polynomial modulo the current modulus. The modulus must be less than 100, and a prime number.

Access: Arithmetic, ↰ ARITH MODULO

Input: The expression to be factorized.

Output: The factorized expression modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Factorize the following expression modulo the default modulus, 3.

x^2+2

Command: `FACTORMOD(X^2+2)`

Result: `(X+1)*(X-1)`

FACTORS

Type: Command

Description: For a value or expression, returns a list of prime factors and their multiplicities.

Access: Arithmetic, ↰ ARITH

Input: A value or expression.

Output: A list of prime factors of the value or expression, with each factor followed by its multiplicity.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example 1: Find the prime factors of 100.

Command: `FACTORS(100)`

Result: `{5 2. 2 2.}`

Example 2: Find the irreducible factors of: $x^2 + 4x + 4$

Command: `FACTORS(X^2+4*X+4)`

Result: `{X+2,2.}`

FCOEF

Type: Command

Description: From an array of roots and multiplicities/poles, returns a rational polynomial with a leading coefficient of 1, with the specified set of roots or poles, and with the specified multiplicities.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: An array of the form [Root 1, multiplicity/pole 1, Root 2, Multiplicity/pole 2, . .] The multiplicity/pole must be an integer. A positive number signifies a multiplicity. A negative number signifies a pole.

Output: The rational polynomial with the specified roots and multiplicities/poles.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the rational polynomial corresponding to the following set of roots and poles:
1, 2, 3, –1

Command: `FCOEF([1,2,3,-1])`

Result: `(X-1)^2/(X-3)`

See also: FROOTS

FOURIER

Type: Function

Description: Returns the n^{th} coefficient of a complex Fourier series expansion. The `PERIOD` variable must be in the current path, and set to hold *L*, the period of the input function.

Access: Calculus ↰ CALC DERIV. & INTEG

Input: Level 1/Argument 2: An expression
Level 2/Argument 1: The number, *n*, of the coefficient to return.

Output: The *n*th Fourier coefficient of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Complex mode must be set, that is, flag –103 must be set.

FROOTS

Type: Command

Description: For a rational polynomial, returns an array of its roots and poles, with their corresponding multiplicities.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: A rational polynomial.

Output: An array of the form [Root 1, Multiplicity 1, Root 2, Multiplicity 2 . . .]
A negative multiplicity indicates a pole.

Flags: Exact mode must be set (flag –105 clear). Numeric mode must not be set (flag –03 clear). If complex mode is set (flag –103 set), FROOTS looks for complex solutions as well as real solutions.

If approximate mode is set (flag –105 set) FROOTS searches for numeric roots.

See also: FCOEF

FXND

Type: Command

Description: Splits an object into a numerator and a denominator.

Access: Catalog, CAT

Input: A fraction, or an object that evaluates to a fraction.

Output: The object split into numerator and denominator.
Level 2/Item 1: The numerator.
Level 1/Item 2: The denominator.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Return the numerator and the denominator of the following expression:

$$\frac{(x-3)^2}{z+4}$$

Command: `FXND((X-3)^2/(Z+4))`

Result: Level 2/Item1: `(X-3)^2`
Level 1/Item 2: `Z+4`

GAUSS

Type: Command

Description: Returns the diagonal representation of a quadratic form.

Access: Matrices, ↰ MATRICES QUADRATIC FORM

Input: Level 2/Argument 1: The quadratic form.
Level 1/Argument 2: A vector containing the independent variables.

Output: Level 4/Argument 1: An array of the coefficients of the diagonal.
Level 3/Argument 2: A matrix, P, such that the quadratic form is represented as P^TDP, where the diagonal matrix D contains the coefficients of the diagonal representation.
Level 2/Argument 3: The diagonal representation of the quadratic form.
Level 1/Argument 4: A list of the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Gaussian symbolic quadratic form of the following:

$x^2 + 2axy$

Command: `GAUSS(X^2+2*A*X*Y,[X,Y])`

Result: `{[1,-A^2] [[1,A][0,1]],-(A^2*Y^2)+(A*Y+X)^2,[X,Y])`

See also: AXQ
QXA

GCD

Type: Function

Description: Returns the greatest common divisor of two objects.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: An expression, or an object that evaluates to a number.
Level 1/Argument 2: An expression, or an object that evaluates to a number.

Output: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Flags: For a symbolic result, clear the CAS modes Numeric option (flag –03 clear).

Example: Find the greatest common divisor of 2805 and 99.

Command: `GCD(2805,99)`

Result: `33`

See also: GCDMOD
EGCD
IEGCD
LCM

GCDMOD

Type: Function

Description: Finds the greatest common divisor of two polynomials modulo the current modulus.

Access: Arithmetic, ↰ (ARITH) MODULO

Input: Level 2/Argument 1: A polynomial expression.
Level 1/Argument 2: A polynomial expression.

Output: The greatest common divisor of the two expressions modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: GCD

HADAMARD

Type: Command

Description: Performs an element by element multiplication of two matrices (Hadamard product).

Access: Matrices, ↰ (MATRICES) OPERATIONS

Input: Level 2/Argument 1: Matrix 1.
Level 1/Argument 2: Matrix 2.
The matrices must have the same order.

Output: The matrix representing the result of the multiplication.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Hadamard product of the following two matrices:

$$\begin{bmatrix} 3 & -1 & 2 \\ 0 & 1 & 4 \end{bmatrix} \text{ and } \begin{bmatrix} 2 & 3 & 0 \\ 1 & 5 & 2 \end{bmatrix}$$

Command: `HADAMARD([[3,-1,2][0,1,4]],[2,3,0][1,5,2]])`

Result: `[[6,-3,-0][0,5,8]]`

HALFTAN

Type: Command

Description: Transforms an expression by replacing sin(x), cos(x) and tan(x) subexpressions with tan($x/2$) terms.

Access: Trigonometry, ↱ TRIG

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

HERMITE

Type: Function

Description: Returns the nth Hermite polynomial.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: A non-negative integer.

Output: The corresponding polynomial expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Hermite polynomial with degree 4.

Command: `HERMITE(4)`

Result: `16*X^4-48*X^2+12`

HESS

Type: Command

Description: Returns the Hessian matrix and the gradient of an expression with respect to the specified variables.

Access: Calculus ↰ CALC DERIV & INTEG

Input: Level 2/Argument 1: An expression.
Level 1/Argument 2: A vector of the variables.

Output: Level 3/Item 1: The Hessian matrix with respect to the specified variables.
Level 2/Item 2: The gradient with respect to the variables.
Level 1/Item 3: The vector of the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: CURL
DIV

HILBERT

Type: Command

Description: Returns a square Hilbert matrix of the specified order.

Access: Matrices, ↰ MATRICES CREATE

Input: A positive integer, representing the order.

Output: The Hilbert matrix of the specified order.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the order 3 Hilbert matrix.

Command: `HILBERT(3)`

Result:

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

HORNER

Type: Command

Description: Executes a Horner scheme on a polynomial. That is, for a given polynomial *P*, and a number *r*, HORNER returns $P/(x-r)$, *r* AND $P(r)$

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: A polynomial, *P*.
Level 1/Argument 2: A number, *r*.

Output: Level 3/Item 1: $P/(x-r)$
Level 2/Item 2: r
Level 1/item 3: $P(r)$, the remainder of the division process.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: For r = 3, find the result of executing a Horner scheme on the following polynomial:

$x^2 + x + 1$

Command: `HORNER(X^2+X+1,3)`

Results: `(X+4,3,13)`

I→R

Type: Command

Description: Converts an integer into a real number.

Access: Catalog, CAT

Input: Level 1/Argument 1: An integer.

Output: Level 1/Item 1: The integer converted to a real number.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: R→I

IABCUV

Type: Command

Description: Returns a solution in integers u and v of $au + bv = c$, where a, b, and c are integers.

Access: Arithmetic, ↰ ARITH INTEGER

Input: Level 3/Argument 1: the value of a.
Level 2/Argument 2: the value of b.
Level 1/Argument 3: the value of c.

Output: Level 2/Item 1: The value for u.
Level 1/Item 2: The value for v.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find a solution in integers of the equation:
$6a + 11b = 3$

Command: `IABCUV(6,11,3)`

Result: `{6,-3}`

See also: ABCUV
IEGCD

IBERNOULLI

Type: Function

Description: Returns the nth Bernoulli number for a given integer.

Access: Catalog, CAT

Input: Level 1/Argument 1: an integer.

Output: Level 1/Item 1: The corresponding nth Bernoulli number for the integer.

Flags: Numeric mode must not be set (flag –03 clear).

IBP

Type: Command

Description: Performs integration by parts on a function. The function must be able to be represented as a product of two functions, where the antiderivative of one of the functions is known:
`f(x) = u(x).v'(x)`

Note that the command is designed for use in RPN mode only.

Access: Calculus, ↰ CALC DERIV & INTEG

Input: Level 2: The integrand expressed as a product of two functions, `u(x).v'(x)`
Level 1: The antiderivative of one of the component functions, v(x).

Output: Level 2: u(*x*)v(*x*)
Level 1: -u'(*x*)v(*x*)

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Use integration by parts to calculate the following:
$\int x\cos(x)dx$

Command 1: Apply the IBP command:
Level 2: `X*COS(X)`
Level 1: `SIN(X)`

Result: Level 2: `SIN(X).X`
Level 1: `-SIN(X)`

Command 2: Apply the INTVX command to level 1, `-SIN(X)`

Result: Level 2: `SIN(X).X`
Level 1: `COS(X)`

Command 3: Press + to add the result to the value at level 2 to obtain the final result.

Result: `SIN(X).(X)+COS(X)`

See also: INTVX, INT, PREVAL, RISCH

ICHINREM

Type: Command

Description: Solves a system of two congruences in integers using the Chinese Remainder theorem.

Access: Arithmetic, ↰ (ARITH) INTEGER

Input: Level 2/Argument 1: A vector of the first value and the modulus.
Level 1/Argument 2: A vector of the second value and the modulus.

Output: A vector of the solution.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following system of congruences:
$x \equiv 2$ Modulo 3
$x \equiv 1$ Modulo 5

Command: `ICHINREM([2,3],[1,5])`

Results: `[-4, 15]`

See also: CHINREM

IDIV2

Type: Command

Description: For two integers, *a* and *b*, returns the integer part of *a*/*b*, and the remainder, *r*.

Access: Arithmetic, ↰ (ARITH) INTEGER

Input: Level 2/Argument 1: *a*.
Level 1/Argument 2: *b*.

Output: Level 2/Item 1: The integer part of *a*/*b*.
Level 1/Item 2: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Return the integer part and the remainder of 11632/864.

Command: `IDIV2(11632,864)`

Result: `{13,400}`

See also: DIV2

IEGCD

Type: Command

Description: Given two integers x and y, returns three integers, a, b, and c, such that:
$ax+by=c$

Access: Arithmetic, ↰ ARITH INTEGER

Input: Level 2/Argument 1: x.
Level 1/Argument 2: y.

Output: Level 3/Item 1: c.
Level 2/Item 2: a.
Level 1/Item 3: b.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: EGCD

ILAP

Type: Function

Description: Returns the inverse Laplace transform of an expression. The expression must evaluate to a rational fraction.

Access: Calculus, ↰ CALC DIFFERENTIAL EQNS

Input: A rational expression.

Output: The inverse Laplace transformation of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the inverse Laplace transform of:

$$\frac{1}{(x-5)^2}$$

Command: `ILAP(1/(X-5)^2)`

Result: `X*EXP(5*X)`

INT

Type: Function

Description: Calculates the antiderivative of a function for a given variable, at a given point.

Access: Catalog, CAT

Input: Level 3/Item 1: A function.
Level 2/Item 2: The variable to obtain the derivative with respect to.
Level 1/Item 3: The point at which to calculate the antiderivative. This point can be a variable or an expression.

Output: The antiderivative of the function for the given variable, at the point you specified.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: INTVX
RISCH

INTVX

Type: Function

Description: Finds the antiderivative of a function symbolically, with respect to the current default variable.

Access: Calculus, ↰ CALC DERIV. & INTEG

Input: An expression.

Output: The antiderivative of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the antiderivative of the following:

$x^2 \ln x$

Command: `INTVX(X^2*LN(X))`

Result: `1/3*X^3*LN(X)+(-1/9)X^3`

See also: IBP, RISCH, PREVAL

INVMOD

Type: Function

Description: Performs modular inversion on an object modulo the current modulus.

Access: Arithmetic, ↰ (ARITH) MODULO

Input: An object.

Output: The modular inverse of the object.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following for *x*, modulo the default modulus, 3.
$(2x \equiv 1)$

Command: `INVMOD(2)`

Result: `-1`

IQUOT

Type: Function

Description: Returns the integer quotient of two integers. That is, given two integers, *a* and *b*, returns the integer *q*, such that:
$a = qb + r$, and $0 \le r < b$

Access: Arithmetic, ↰ (ARITH) INTEGER

Input: Level 2/Item 1: The dividend.
Level 1/Item 2: The divisor.

Output: The integer quotient.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: QUOT
IDIV2

IREMAINDER

Type: Function

Description: Returns the remainder of an integer division.

Access: Catalog, CAT

Input: Level 2/Argument 1: The numerator.
Level 1/Argument 2: The denominator.

Output: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: IDIV2

ISPRIME?

Type: Function

Description: Tests if a number is prime.

Access: Arithmetic, ↰ ARITH INTEGER

Input: An object that evaluates to a number.

Output: 1 (True) if the number is prime, 0 (False) if it is not.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: NEXTPRIME
PREVPRIME

JORDAN

Type: Command

Description: Computes the eigenvalues, eigenvectors, minimum polynomial, and characteristic polynomial of a matrix.

Access: Matrices, ↰ MATRICES EIGENVECTORS

Input: An $n \times n$ matrix.

Output: Level 4/Item 1: The minimum polynomial.
Level 3/Item 2: The characteristic polynomial.
Level 2/Item 3: A list of characteristic spaces tagged by the corresponding eigenvalue (either a vector or a list of Jordan chains, each of them ending with an "Eigen:"-tagged eigenvector).
Level 1/Item 4: An array of the eigenvalues, with multiplicities

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LAGRANGE

Type: Command

Description: Returns the interpolating polynomial of minimum degree for a pair of values.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: A two × *n* matrix of the *n* pairs of values.

Output: The polynomial that results from the Lagrange interpolation of the data.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find an interpolating polynomial for the data (1,6), (3,7), (4,8), (2,9)

Command: LAGRANGE($\begin{bmatrix} 1 & 3 & 4 & 2 \\ 6 & 7 & 8 & 9 \end{bmatrix}$)

Result: $\frac{8x^3 - 63x^2 + 151x - 60}{6}$

LAP

Type: Function

Description: Performs a Laplace transform on an expression with respect to the current default variable.

Access: Calculus, ↰ CALC DIFFERENTIAL EQNS

Input: An expression.

Output: The Laplace transform of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Laplace transform of e^x.

Command: `LAP(EXP(X))`

Result: `1/(X-1)`

LAPL

Type: Command

Description: Returns the Laplacian of a function with respect to a list of variables.

Access: ↰ CALC DERIV & INTEG

Input: Level 2/Argument 1:An expression.
Level 1/Argument 2: A vector of variables.

Output: The Laplacian of the expression with respect to the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find, and simplify, the Laplacian of the following expression:

$e^x \cos(zy)$

Command:
```
LAPL(EXP(X)*COS(Z*Y),[X,Y,Z])
EXPAND(ANS(1))
```

Result: `-((Y^2+Z^2-1)*EXP(X)*COS(Z*Y))`

LCM

Type: Function

Description: Returns the least common multiple of two objects.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: An expression, a number, or object that evaluates to a number.
Level 1/Argument 2: An expression, a number, or object that evaluates to a number.

Output: The least common multiple of the objects.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the least common multiple of the following two expressions:

$x^2 - 1$

$x-1$

Command: `LCM(X^2-1,X-1)`

Results: `X^2-1`

See also: GCD

LCXM

Type: Command

Description: From a program with two arguments, builds a matrix with the specified number of rows and columns, with $a_{ij} = f(i,j)$.

Access: Catalog, CAT

Input: Level 3/Argument 1: The number of rows you want in the resulting matrix.
Level 2/Argument 2: The number of columns you want in the resulting matrix.
Level 1/Argument 3: A program that uses two arguments.

Output: The resulting matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Build a 2 × 3 matrix with a_{ij}=i+2j.

Command: `LCXM(2,3,<<→I J 'I+2*J'>>)`

Result:

$$\begin{bmatrix} 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}$$

LDEC

Type: Command

Description: Solves a linear differential equation with constant coefficients, or a system of first order linear differential equations with constant coefficients.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: For a single equation, the function forming the right hand side of the equation. For a system of equations, an array comprising the terms not containing the dependent variables.
Level 1/Argument 2: For one equation, the auxiliary polynomial. For a system of equations, the matrix of coefficients of the dependent variables.

Output: The solution.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LEGENDRE

Type: Function

Description: Returns the *n*th degree Legendre polynomial.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: An integer, *n*.

Output: The *n*th Legendre polynomial.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Legendre polynomial with degree 4.

Command: `LEGENDRE(4)`

Result: `(35*X^4-30*X^2+3)/8`

LGCD

Type: Function

Description: Returns the greatest common divisor of a list of expressions or values.

Access: Arithmetic, ↰ ARITH

Input: A list of expressions or values.

Output: Level 2/Item 1: The list of elements.
Level 1/Item 2: The greatest common divisor of the elements.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LIMIT

Type: Function

Description: Returns the limit of a function as it approaches a specified value.

Access: Calculus, ↰ CALC LIMITS&SERIES

Input: Level 2/Argument 1: An expression.

Level 1/Argument 2: An expression of the form $x = y$, where x is the variable and y is the value at which the limit is to be evaluated.

Output: The limit of the expression at the limit point.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the following limit:

$$\lim_{x \to y} \frac{x^n - y^n}{x - y}$$

Command: `LIMIT((X^N-Y^N)/(X-Y), X=Y)`

Result: `N*EXP(N*LN(Y))/Y`

See also: SERIES

LIN

Type: Command

Description: Linearizes expressions involving exponential terms.

Access: Exponential and logarithm, ↰ EXP&LN

Input: An expression.

Output: The linearized expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Linearize the following expression:

$x(e^x e^y)^4$

Command: `LIN(X*(EXP(X)*EXP(Y))^4)`

Result: `X*EXP(4X+4Y)`

LINSOLVE

Type: Command

Description: Solves a system of linear equations.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: An array of equations.
Level1/Argument 2: A vector of the variables to solve for.

Output: Level 3/Item 1: The system of equations.
Level 2/Item 2: A list of the pivot points.
Level 1/Item 3: The solution.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LNAME

Type: Command

Description: Returns the variable names contained in a symbolic expression.

Access: Catalog, CAT

Input: A symbolic expression.

Output: A vector containing the variable names.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: LVAR

LNCOLLECT

Type: Command

Description: Simplifies an expression by collecting logarithmic terms.

Access: Algebra, ↱ ALG

Input: An expression.

Output: The simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:
$2(\ln(x)+\ln(y))$

Command: `LNCOLLECT(2(LN(X)+LN(Y))`

Result: `LN(X^2+Y)`

LVAR

Type: Command

Description: Returns a list of variables in an algebraic object.

Access: Catalog, CAT

Input: An algebraic object.

Output: Level 2/Item 1: The algebraic object.
Level 1/Item 2: A vector of the variables that the object contains.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

MAD

Type: Command

Description: Returns details of a square matrix.

Access: Matrices, ↰ MATRICES OPERATIONS

Input: A square matrix

Output: Level 4/Item 1: The determinant.
Level 3/Item 2: The formal inverse.
Level 2/Item 3: The matrix coefficients of the polynomial, p, defined by $(xi–a)p(x)=m(x)i$, where a is the matrix, and m is the characteristic polynomial of a.
Level 1/Item 4: The characteristic polynomial.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

MENUXY

Type: Command

Description: Displays a function key menu of the computer algebra commands in the specified range.

Access: Catalog, CAT

Input: Level 2/Argument 1: The number of the first command in the range that you want to display.
Level 1/Argument 2: The number of the last command in the range that you want to display.

Output: On the function key menu, a list of the computer algebra commands in the range that you specified.

MODSTO

Type: Command

Description: Changes the modulo setting to the specified number. The number that you set is reflected in the CAS Modes input form.

Access: Arithmetic, ↰ ARITH MODULO

Input: The modulo value that you want to set.

Output: The modulo setting is changed to the specified number.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

MULTMOD

Type: Function

Description: Performs modular multiplication of two objects, modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: A number or an expression.
Level 1/Argument 2: A number or an expression.

Output: The result of modular multiplication of the two objects, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the product of $2x$ and $38x^2$, modulo the default modulus, 3.

Command: `MULTMOD(2*X,38*X^2)`

Result: `X^3`

NEXTPRIME

Type: Function

Description: Given an integer, returns the next prime number larger than the integer.

Access: Arithmetic, ↰ ARITH INTEGER

Input: An integer.

Output: The next prime number larger than the integer.

Example: Find the closest, larger prime number to 145.

Command: `NEXTPRIME(145)`

Result: `149`

See also: ISPRIME?
PREVPRIME

PA2B2

Type: Command

Description: Takes a prime number, p, such that $p=2$ or $p \equiv 1 \mod 4$, and returns a Gaussian integer $a + ib$ such that $p = a^2 + b^2$. This function is useful for factorizing Gaussian integers.

Access: Arithmetic, ↰ ARITH INTEGER

Input: A prime number, p, such that $p=2$ or $p \equiv 1 \mod 4$

Output: A Gaussian integer a+ib such that $p=a^2+b^2$

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: GAUSS

PARTFRAC

Type: Command

Description: Performs partial fraction decomposition on a partial fraction.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: An algebraic expression.

Output: The partial fraction decomposition of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Perform a partial fraction decomposition of the following expression:

$$\frac{1}{x^2-1}$$

Command: `PARTFRAC(1/(X^2-1))`

Result: `(-1+(1/X+^2))`

PCAR

Type: Command

Description: Returns the characteristic polynomial of an $n \times n$ matrix.

Access: Matrices, ↰ MATRICES EIGENVECTORS

Input: A square matrix.

Output: The characteristic polynomial of the matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the characteristic polynomial of the following matrix:

$$\begin{bmatrix} 5 & 8 & 16 \\ 4 & 1 & 8 \\ -4 & -4 & -11 \end{bmatrix}$$

Command: `PCAR([[5,8,16][4,1,8][-4,-4,-11]])`

Result: `X^3+5*X^2+3*X-9`

POWMOD

Type: Function

Description: Raises an object (number or expression) to the specified power, and expresses the result modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The object.
Level 1/Argument 2: The exponent.

Output: The result of the object raised to the exponent, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

PREVAL

Type: Function

Description: With respect to the current default variable, returns the difference between the values of a function at two specified values of the variable.

PREVAL can be used in conjunction with INTVX to evaluate definite integrals. See the example below.

Access: Calculus, ↰ CALC DERIV. & INTEG.

Input: Level 3/Argument 1: A function.
Level 2/Argument 2: The lower bound.
Level 3/Argument 1: The upper bound.
The bounds can be expressions.

Output: The result of the evaluation.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Evaluate the following:

$$\int_0^3 (x^3 + 3x)dx$$

Command: `PREVAL(INTVX(X^3+3*X),0,3)`

Result: `135/4`

PREVPRIME

Type: Function

Description: Given an integer, finds the closest prime number smaller than the integer.

Access: Arithmetic, ↰ ARITH INTEGER

Input: An integer.

Output: The closest prime number smaller than the integer.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the closest, smaller prime number to 145.

Command: `PREVPRIME(145)`

Result: `139`

See also: ISPRIME?
NEXTPRIME

PROPFRAC

Type: Command

Description: Splits an improper fraction into an integer part and a fraction part.

Access: Arithmetic, ↰ ARITH

Input: An improper fraction, or an object that evaluates to an improper fraction.

Output: A proper fraction.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Express the following as a proper fraction:

$$\frac{x^3+4}{x^2}$$

Command: `PROPFRAC((X^3+4)/X^2))`

Result: `X+(4/X^2)`

PSI

Type: Function

Description: Calculates the polygamma function in one point.

Access: Catalog, CAT

Input: A complex expression.

Output: The polygamma function.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Psi

Type: Function

Description: Calculates the digamma function in one point. The digamma function is the derivative of the natural logarithm (ln) of the gamma function. The function can be represented as follows:

$$\Psi(z) = \frac{d}{dz}(\ln\Gamma(z)) = \frac{\Gamma'(z)}{\Gamma(z)}$$

Access: Catalog, CAT

Input: Level 2/Argument 1: A complex expression
Level 1/Argument 2: A non-negative integer.

Output: The digamma function at the specified point.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

PTAYL

Type: Function

Description: Returns the Taylor polynomial at $x = a$ for a specified polynomial.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: A polynomial, P.
Level 1/Argument 2: A number, *a*.

Output: A polynomial, Q such that $Q(x-a)=P(x)$.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

QUOT

Type: Function

Description: Returns the quotient part of the Euclidean division of two polynomials.

Access: Arithmetic, ↰ (ARITH) POLYNOMIAL

Input: Level 2/Argument 1: The numerator polynomial.
Level 1/Argument 2: The denominator polynomial.

Output: The quotient of the Euclidean division.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the quotient of the division of $x^3 + 6x^2 + 11x + 6$ by $x^2 + 5x + 6$.

Command: `QUOT(X^3+6*X^2+11*X+6, X^2+5*X+6)`

Result: `X+1`

See also: REMAINDER

QXA

Type: Command

Description: Expresses a quadratic form in matrix form.

Access: Catalog, (CAT)

Input: Level 2/Argument 1: A quadratic form.
Level 1/Argument 2: A vector containing the variables.

Output: Level 2/Item 1: The quadratic form expressed in matrix form.
Level 1/Item 2: A vector containing the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Express the following quadratic form in matrix form:
$x^2 + xy + y^2$

Command: `QXA(X^2+X*Y+Y^2, [X,Y])`

Result: `{[[1,1/2][1/2,1]],[X,Y]}`

See also: AXQ
GAUSS

R→I

Type: Command

Description: Converts a real number to an integer.

Access: Catalog, CAT

Input: Level 1/Argument 1: A real number.

Output: Level 1/Item 1: The real number converted to an integer.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: I→R

REF

Type: Command

Description: Reduces a matrix to echelon form.

Access: Matrices, ↰ MATRICES LINEAR SYSTEMS

Input: A matrix.

Output: The equivalent matrix in echelon form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: RREF

REMAINDER

Type: Function

Description: Returns the remainder of the Euclidean division of two polynomials.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: The numerator polynomial.
Level 1/Argument 2: The denominator polynomial.

Output: The remainder resulting from the Euclidean division.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: QUOT

REORDER

Type: Function

Description: Given a polynomial expression and a variable, reorders the variables in the expression in the order of powers set on the CAS Modes screen, that is, either in increasing or decreasing order.

Access: Catalog, CAT

Input: Level 2/Argument 1: The polynomial expression.
Level 1/Argument 2: The variable with respect to which the reordering is performed.

Output: The reordered expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

RESULTANT

Type: Function

Description: Returns the resultant of two polynomials of the current variable. That is, it returns the determinant of the Sylvester matrices of the two polynomials.

Access: Catalog, CAT

Input: Level 2/Argument 1: The first polynomial.
Level 1/Argument 2: The second polynomial.

Output: The determinent of the two matrices that correspond to the polynomials.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

RISCH

Type: Function

Description: Performs symbolic integration on a function using the Risch algorithm. RISCH is similar to the INTVX command, except that it allows you to specify the variable of integration.

Access: Calculus ↰ CALC DERIV. & INTEG.

Input: Level 2/Argument 1: The function to integrate.
Level 1/Argument 2: The variable of integration.

Output: The antiderivative of the function with respect to the variable.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the antiderivative of the following function, with respect to *y*:

$y^2 + 3y + 2$

Command: `RISCH(Y^3-3*Y+2,Y)`

Result: `1/3*Y^3+3/2*Y^2+2*Y`

See also: IBP
INT
INTVX

RREF

Type: Command

Description: Reduces a martrix to row-reduced echelon form.

Access: Matrices, ↰ MATRICES LINEAR SYSTEMS

Input: A matrix.

Output: An equivalent matrix in row reduced echelon form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

---

rref

Type: Command

Description: Reduces a martrix to row-reduced echelon form.

Access: Matrices, ↰ (MATRICES) LINEAR SYSTEMS

Input: A matrix.

Output: Level 2/Argument 1: The pivot points.

Level 1/Argument 2: An equivalent matrix in row reduced echelon form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

RREFMOD

Type: Command

Description: Performs modular row-reduction to echelon form on a matrix, modulo the current modulus.

Access: Catalog, (CAT)

Input: A matrix.

Output: The modular row-reduced matrix. The modulo value is set using the Modes CAS input form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SERIES

Type: Command

Description: For a given function, computes Taylor series, asymptotic development and limit at finite and infinite points.

Access: Calculus, ↰ (CALC) LIMITS & SERIES

Input: Level 3/Argument 1: The function $f(x)$
Level 2/Argument 2: The variable if the limit point is 0, or an equation $x = a$ if the limit point is a.
Level 1/Argument 3: The order for the series expansion. Note the following points:

- The minimum value is 2, and the maximum value is 20.
- If the order is a positive or negative real number, the series is unidirectional.
- For bidirectional series expansions, you need to give the order as a binary integer, for example, #5d.

Output: Level 2/Item 1: A list containing the bidirectional limit, an expression approximating the function near the limit point, and the order of the remainder. These are expressed in terms of a small parameter h.
Level 1/Item 2: An expression for h in terms of the original variable.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SEVAL

Type: Function

Description: In the given expression, evaluates any existing variables that the expression contains and substitutes these back into the expression.

Access: Catalog, CAT

Input: Level 1/Item 1: An algebraic expression.

Output: The expression with existing variables evaluated.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SIGMA

Type: Function

Description: Calculates the discrete antiderivative of a function with respect to the variable that you define.

Access: Catalog, CAT

Input: Level 2/Item 1: A function
Level 1/Item 2: The variable to calculate the antiderivative with respect to.

Output: The antiderivative of the function.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: SIGMAVX, RISCH

SIGMAVX

Type: Function

Description: Calculates the discrete antiderivative of a function with respect to the current variable.

Access: Catalog, CAT

Input: Level 2/Item 1: A function.

Output: The antiderivative of the function.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: SIGMA, RISCH

SIGNTAB

Type: Command

Description: Tabulates the sign of a rational function of one variable.

Access: Catalog, CAT

Input: An algebraic expression.

Output: A list containing, the points where the expression changes sign, and for each point, the sign of the expression between the points.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SIMP2

Type: Command

Description: Simplifies two objects by dividing them by their greatest common divisor.

Access: Arithmetic, ↰ ARITH

Input: Level 2/Argument 1: The first object.
Level 1/Argument 2: The second object.

Output: Level 2/Item 1: The first object divided by the greatest common divisor.
Level 1/Item 2: The second object divided by the greatest common divisor.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Divide the following expressions by their greatest common divisor:

$x^3 + 6x^2 + 11x + 6$

$x^3 - 7x - 6$

Command: `SIMP2(X^3+6*X^2+11*X+6, X^3-7*X-6)`

Result: `{X+3,X-3}`

SINCOS

Type: Command

Description: Converts complex logarithmic and exponential expressions to expressions with trigonometric terms.

Access: Trigonometry, ↱ TRIG

Input: An expression with complex linear and exponential terms.

Output: The expression with logarithmic and exponential subexpressions converted to trigonometric and inverse trigonometric expressions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Must be in complex mode (flag –103 set).

Example: Express e^{ix} in trigonometric terms.

Command: `SINCOS(EXP(i*X))`

Result: `COS(X)+iSIN(X)`

SOLVE

Type: Command

Description: Finds zeros of an expression equated to 0, or solves an equation.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: The expression or equation.
Level 1/Argument 2: The variable to solve for.

Output: A list of zeros or solutions.

Flags: If exact mode is set (flag –105 clear) and there are no exact solutions, the command returns a null list even when there are approximate solutions.

Example: Find the zeros of the following expression:

$x^3 - x - 9$

Command: `SOLVE(X^3-X-9,X)`

Result: `{X=2.24004098747}`

See also: SOLVEVX

SOLVEVX

Type: Command

Description: Finds zeros of an expression with respect to the current variable, or solves an equation with respect to the current variable. (You use the CAS modes input form to set the current variable.)

Access: Symbolic solve, ↰ S.SLV

Input: A function or equation in the current variable.

Output: A list of zeros or solutions.

Flags: For a symbolic result, clear the CAS modes Numeric option (flag –03 clear).
If Exact mode is set (flag –105 clear) and there are no exact solutions, the command returns a null list even when there are approximate solutions.

Example: Solve the following expression for 0, where X is the default variable on the calculator:

$x^3 - x - 9$

Command: `SOLVEVX(X^3-X-9)`

Result: `{X=2.2400}`
Note that if exact mode is set, this example returns a null list as there are no exact solutions to the equation.

See also: SOLVE

SUBST

Type: Function

Description: Substitutes a value for a variable in an expression. The value can be numeric or an expression.

Access: Algebra, ↱ ALG

Input: Level 2/Argument 1: An expression.
Level 1/Argument 2: The value or expression to be substituted.

Output: The expression with the substitution made.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Substitute $x = z+1$ for x in the following expression, and apply the EXPAND command to simplify the result:

$x^2 + 3x + 7$

Command:
```
SUBST(X^2+3*X+7,X=Z+1)
EXPAND(ANS(1))
```

Result: `Z^2+5*Z+11`

SUBTMOD

Type: Function

Description: Performs a subtraction, modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The object or number to be subtracted from.
Level 1/Argument 2: The object or number to subtract.

Output: The result of the subtraction, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SYLVESTER

Type: Command

Description: For a symmetric matrix A, returns D and P where D is a diagonal matrix and $A = P^T DP$

Access: Catalog, CAT

Input: A symmetric matrix.

Output: Level 2/Item 1: the diagonal martrix, D.
Level 1 1/Item 2: The matrix P.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TABVAL

Type: Command

Description: For an expression and a list of values, returns the results of substituting the values for the default variable in the expression.

Access: Catalog, CAT

Input: Level 2/Argument 1: An algebraic expression in terms of the current variable.
Level 1/Argument 2: A list of values for which the expression is to be evaluated.

Output: Level 2/Item 1: The algebraic expression.
Level 1/Item 2: A list containing two lists: a list of the values and a list of the corresponding results.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Substitute 1, 2, and 3 into $x^2 + 1$.

Command: `TABVAL(X^2+1,{1 2 3})`

Result: `{{1 2 3},{2 5 10}}`

TABVAR

Type: Command

Description: For a rational function of the current variable, computes the variation table, that is the turning points of the function and where the function is increasing or decreasing.

Access: Catalog, CAT

Input: A rational function of the current variable.

Output: Level 3/Item 1: The original rational function.
Level 2/Item 2: A list of two lists. The first list indicates the variation of the function (where it is increasing or decreasing) in terms of the independent variable. The second list indicates the variation in terms of the dependent variable.
Level 1/Item 3: A graphic object that shows how the variation table was computed.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TAN2SC

Type: Command

Description: Replaces tan(*x*) sub-expressions with sin(*x*)/(1-cos(2*x*) or (1-cos(2*x*))/sin(2*x*).

Access: Trigonometry, ↱ TRIG

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TAN2SC2

Type: Command

Description: Replaces tan(*x*) terms in an expression with sin(2*x*)/1+cos(2*x*) terms.

Access: Trigonometry, ⮡ TRIG

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Flag –116 – Prefer sin/cos affects the result:

- If flag –116 is set (Prefer sin()) then Tan(x) terms are replaced with:
  1 – Cos(2*x*)/Sin(2*x*)
- If flag –116 is clear (prefer Cos()) then Tan(*x*) terms are replaced with:
  Sin(2*x*)/1 + Cos(2*x*)

TAYLOR0

Type: Function

Description: Performs a fourth-order Taylor expansion of an expression at $x = 0$.

Access: Calculus, ⮢ CALC LIMITS & SERIES

Input: An expression

Output: The Taylor expansion of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TCHEBYCHEFF

Type: Function

Description: Returns the nth Tchebycheff polynomial.

Access: Catalog, CAT

Input: A non-negative integer, *n*.

Output: The *n*th Tchebycheff polynomial.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TCOLLECT

Type: Command

Description: Linearizes products in a trigonometric expression by collecting sine and cosine terms, and by combining sine and cosine terms of the same argument.

Access: Trigonometry, → TRIG

Input: An expression with trigonometric terms.

Output: The simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TEXPAND

Type: Command

Description: Expands transcendental functions.

Access: Trigonometry, ← TRIG

Input: An expression.

Output: The transformation of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:
ln(sin(*x*+*y*))

Command: `TEXPAND(LN(SIN(X+Y)))`

Result: `LN(COS(Y)*SIN(X)+SIN(Y)* COS(X))`

TLIN

Type: Command

Description: Linearizes and simplifies trigonometric expressions. Note that this function does not collect sin and cos terms of the same angle.

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformation of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Linearize and simplify the following:
$(\cos(x))^4$

Command: `TLIN(COS(X)^4)`

Result: `(1/8)*COS(4X)+(1/2)*COS(2X)+(3/8)`

TRAN

Type: Command

Description: Returns the transpose of a matrix.

Access: Matrices, ↰ MATRICES OPERATIONS

Input: A matrix.

Output: The transposed matrix.

Flags: Exact mode must be set (flag –105 clear). Numeric mode must not be set (flag –03 clear).

Example: Transpose the following matrix: $\begin{bmatrix} 1 & 7 \\ 2 & -3 \end{bmatrix}$

Command: `TRAN([[1,7][2,-3]])`

Result: `[[1,2][7,-3]]`

TRIG

Type: Command

Description: Converts complex logarithmic and exponential subexpressions into their equivalent trigonometric expressions.

Access: Trigonometry, ↱ TRIG

Input: A complex expression with logarithmic and/or exponential terms.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Must be in Complex mode (flag –103 set).

Example: Express the following in trigonometric terms:
$\ln(x + i)$

Command: `TRIG(LN(X+i))`

Result: $\dfrac{\mathrm{LN}(X^2 + 1) + 2 \bullet i \bullet \mathrm{ATAN}\left(\frac{1}{x}\right)}{2}$

TRIGCOS

Type: Command

Description: Simplifies a trigonometric expression by applying the identity:
$(\sin x)^2 + (\cos x)^2 = 1$
Returns only cosine terms if possible.

Access: Trigonometry, ↱ TRIG

Input: An expression with trigonometric terms.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TRIGSIN, TRIGTAN

TRIGSIN

Type: Command

Description: Simplifies a trigonometric expression by applying the identity:
$(\sin x)^2 + (\cos x)^2 = 1$
Returns only sine terms if possible.

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TRIGCOS, TRIGTAN

TRIGTAN

Type: Command

Description: Replaces sin and cos terms in a trigonometric expression with tan terms.

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TRIGCOS, TRIGSIN

TRUNC

Type: Command

Description: Truncates a series expansion.

Access: Catalog, CAT

Input: Level 2/Argument 1: The expression that you want to truncate.
Level 1/Argument 2: The expression to truncate with respect to.

Output: The expression from Level 2/Argument 1, with terms of order greater than or equal to the order of th expression in Level 1/Argument 2 removed.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TSIMP

Type: Command

Description: Performs simplifications on expressions involving exponentials and logarithms.

Access: Exponential and logarithms, ↰ EXP&LN

Input: An expression

Output: The simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TEXPAND
TLIN

VANDERMONDE

Type: Command

Description: Builds a Vandermonde matrix from a list of objects. That is, for a list of n objects, the command creates an $n \times n$ matrix. The i^{th} row in the matrix consists of the list items raised to the power of $(i–1)$.

Access: Matrices, ↰ MATRICES CREATE

Input: A list of objects.

Output: The corresponding Vandermonde matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Build a Vandermonde matrix from the following list of objects:
$\{x, y, z\}$

Command: `VANDERMONDE({x,y,z})`

Result:

$$\begin{bmatrix} 1 & 1 & 1 \\ x & y & z \\ x^2 & y^2 & z^2 \end{bmatrix}$$

VER

Type: Command

Description: Returns the Computer Algebra System version number, and date of release.

Access: Catalog, CAT

Input: No input required.

Output: The version and release date of the Computer Algebra System software.

XNUM

Type: Command

Description: Converts an object or a list of objects to approximate numeric format.

Access: Catalog, CAT

Input: An object or list of objects.

Output: The objects in numeric format.

Example: Find the approximate value of $\pi/2$, $3e$, and $4\cos(2)$.

Command: `XNUM({π/2,3*e,4*COS(2})`

Results: `{1.5707963268 8.15484548538 -1.66458734619}`

XQ

Type: Command

Description: Converts a number, or a list of numbers in decimal format, to rational format.

Access: Catalog, CAT

Input: A number, or a list of numbers.

Output: The number or list of numbers in rational format.

Example: Express .3658 in rational format:

Command: `XQ(.3658)`

Results: `1829/5000`

ZEROS

Type: Command

Description: Returns the zeros of a function of one variable, without multiplicity.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: An expression.
Level 1/Argument 2: The variable to solve for.

Output: The solution or solutions for the expression equated to 0.

Flags: For a symbolic result, clear the CAS modes Numeric option (flag –03 clear).
The following flag –settings affect the result:

- If Exact mode is set (flag –105 is clear), the function attempts to find exact solutions only. This may return a null list, even if approximate solutions exist.
- If Approximate mode is set (flag –105 set), the function finds numeric roots.
- If Complex mode is set flag –103 set, the function searches for real and complex roots.
