Chapter 7

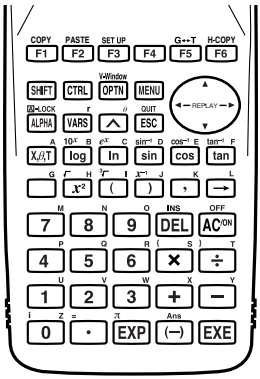
7

Computer Algebra System and Tutorial Modes (ALGEBRA FX 2.0 PLUS only)

7-1 Using the CAS (Computer Algebra System) Mode

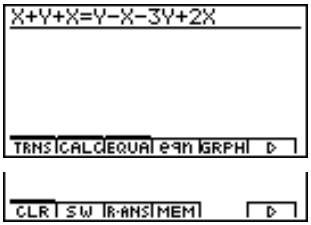
On the Main Menu, select the **CAS** icon to enter the CAS Mode.

The following table shows the keys that can be used in the CAS Mode.

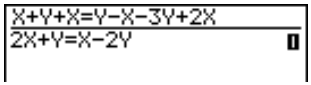


■ Inputting and Displaying Data

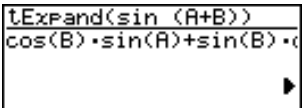
Input in the Algebra Mode is performed in the upper part of the display, which is called the "input area." You can input commands and expressions at the current cursor location.



Calculation results appear in the lower part of the display, which is called the "output area." When a calculation produces an equation or inequality, the lower part of the display is divided between a "natural result display area" for the result, and a "formula number area" for the formula number as shown below.



If all the result does not fit on the display, use the cursor keys to scroll it.



■ Inputting List Data

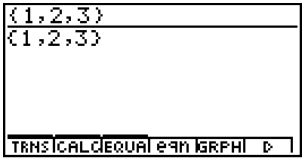
List: {element, element, ..., element}

- Elements should be separated by commas, and the entire set of elements should be enclosed within {curly braces}.
- You can input numeric values and expressions, equations, and inequalities as list elements.

● ● ● ● ●

Example **To input List {1, 2, 3}**

SHIFT × ({) 1 , 2 , 3
SHIFT ÷ (}) EXE



■ Inputting Matrix Data

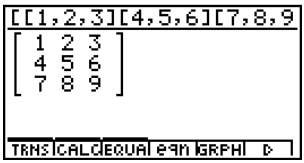
Matrix ($m \times n$): [[(1,1) entry, (1,2) entry, ..., (1,m) entry] [(2,1) entry, ..., (2,n) entry]...
[(m, n) entry, ..., (m, n) entry]]

- The above input is arranged to show the relative positions of entries in the matrix. Actual input is an unbroken line, from left to right.
- Entries should be separated by commas, and the entire set of elements should be enclosed within [square brackets]. And each line also should be enclosed within [square brackets].
- You can input numeric values and expressions as matrix entries.

● ● ● ● ●

Example **To input the matrix shown below** $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

SHIFT + ([ ) SHIFT + ([ ) 1 , 2 , 3
SHIFT − ( ]) SHIFT + ([ ) 4 , 5 , 6
SHIFT − ( ]) SHIFT + ([ ) 7 , 8 , 9
SHIFT − ( ]) SHIFT − ( ]) EXE



■ Inputting Vector Data

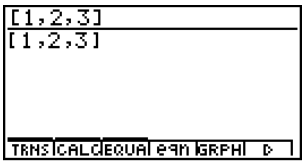
Vector: [component, component, ..., component]

- Components should be separated by commas, and the entire set of components should be enclosed within [square brackets].
- You can input numeric values and expressions as vector component entries.

● ● ● ● ●

Example **To input Vector (1 2 3)**

SHIFT ⊞ ([ ) 1 , 2 , 3
SHIFT ⊟ ( ]) EXE



■ Performing an Algebra Mode Operation

There are two methods that you can use for input in the Algebra Mode.

- Function menu command input
- Manual formula and parameter input

■ Menu Command Input

Press a function menu key to display the menu of functions for the type of operation you are trying to perform.

- **TRNS** ... {formula transformation menu}
- **CALC** ... {formula calculation menu}
- **EQUA** ... {equation, inequality menu}
- **eqn** ... {calls up an equation stored in Equation Memory in accordance with a specified input value}
- **CLR** ... {variable/formula delete menu}

Pressing the OPTN key displays the menu shown below.

- **LIST** ... {list calculation menu}
- **MAT** ... {matrix calculation menu}
- **VECT** ... {vector calculation menu}

For details on commands and their formats, see the “Algebra Command Reference” on page 7-1-11.

■ Manual Formula and Parameter Input

You can use the function menus, [OPTN] key, and [VARS] key in combination to input formulas and parameters as described below.

- [F3](EQUA)[1](INEQUA)
  - {**>**}/{**<**}/{**≥**}/{**≤**} ... {inequality}
- [OPTN] key
  - {∞}/{**Abs**}/{x**!**}/{**sign**} ... {infinity}/{absolute value}/{factorial}/{signum function*1}
  - {**HYP**} ... {hyperbolic}/{inverse hyperbolic} functions
    - **{sinh}/{cosh}/{tanh}/{sinh^{-1}}/{cosh^{-1}}/{tanh^{-1}}**
- [VARS] key
  - {**Yn**}/{**rn**}/{**Xtn**}/{**Ytn**}/{**Xn**} ... input of graph memory {Yn}/{rn}/{Xtn}/{Ytn}/{Xn}

■ Formula Memory

The CAS Mode has 28 formula variables. Variable names are the letters A through Z, plus r, and θ. CAS Mode formula variables are independent of standard value variables.

● ● ● ● ●

Example **To assign a formula that differentiates sin(X) at X (cos(X)) to variable A**

[F2](CALC)[1](diff)[sin][X,θ,T][,]
[X,θ,T][)][→][ALPHA][X,θ,T](A)[EXE]

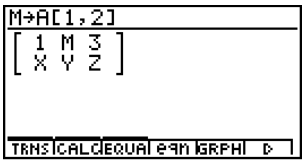
```
diff(sin X,X)→A
cos(X)
```

*1 $signum\ (A)\begin{cases} 1 \text{ (real number, } A > 0) \\ -1 \text{ (real number, } A < 0) \\ \dfrac{A}{|A|} (A = \text{imaginary number}) \\ \text{Undefined } (A = 0) \end{cases}$

● ● ● ● ●

Example **To assign M to row 1 column 2 of variable A when the matrix $\begin{bmatrix} 1 & 2 & 3 \\ X & Y & Z \end{bmatrix}$ is assigned to it**

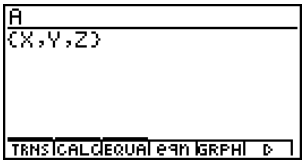
[ALPHA] [7] (M) [→] [ALPHA] [X,θ,T] (A)
[SHIFT] [+] ([ ) [1] [,] [2] [SHIFT] [−] ( ]) [EXE]



● ● ● ● ●

Example **To recall the value of variable A when the list {X, Y, Z} is assigned to it**

[ALPHA] [X,θ,T] (A) [EXE]



● ● ● ● ●

Example **To recall the first component (A [1]) of variable A when vector (X Y Z) is assigned to it**

[ALPHA] [X,θ,T] (A) [SHIFT] [+] ([ ) [1]
[SHIFT] [−] ( ]) [EXE]

■ Function Memory and Graph Memory

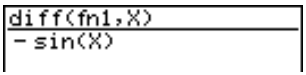
Function memory lets you store functions for later recall when you need them.

With graph memory, you can store graphs in memory. Press the [VARS] key and then input the name of the graph.

● ● ● ● ●

Example **To differentiate $f_1 = \cos(X)$, which is assigned to function memory f_1, at X**

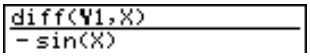
[F2] (CALC) [1] (diff) [OPTN] [F6] (▷) [F4] (FMEM)
[3] (fn) [1] [,] [X,θ,T] [)] [EXE]



● ● ● ● ●

Example **To differentiate Y1 = cos(X), which is assigned to graph memory Y1, at X**

[F2] (CALC) [1] (diff)
[VARS] [F1] (Yn) [1] [,] [X,θ,T] [)] [EXE]



■ Eqn Memory

When a calculation result is an equation or inequality, its formula number is displayed in the formula number area, and the equation is stored in Eqn memory.*1 Stored equations can be recalled with the eqn command, rclEqn command or rclAllEqn command.

*1 Up to 99 formulas can be stored in Eqn memory.
The error message “Memory ERROR” when you try to store an equation when there are already 99 equations in Eqn memory. When this happens, execute the ALLEQU (Delete All Equations) from the CLR menu.

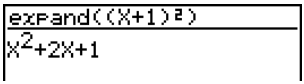
■ Answer (Ans) Memory and Continuous Calculation

Answer (Ans) memory and continuous calculation can be used just as with standard calculations. In the Algebra Mode, you can even store formulas in Ans memory.

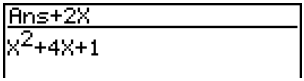
● ● ● ● ●

Example **To expand $(X+1)^2$ and add the result to 2X**

[F1] (TRNS) [1] (expand)
[(] [X,θ,T] [+] [1] [)] [x^2] [)] [EXE]



Continuing:
[+] [2] [X,θ,T] [EXE]



■ Replay Contents

Replay memory can be used in the input area. After a calculation is complete, pressing ◀ or ▶ in the input area recalls the formula of the last calculation performed. After a calculation or after pressing [AC], you can press ▲ or ▼ to recall previous formulas.

■ Moving the Cursor Between Display Areas

When ◄ ► ▲ ▼ indicates a calculation result that does not fit on the display, the cursor keys perform output area scrolling. To use the Replay Function from this condition, press [F6] (▷) [F2] (SW). ◄ ► ▲ ▼ change to a dotted line display to indicate that cursor key operations control the input area.

Pressing [F2] (SW) again moves the cursor back to the output area.

Pressing [F6] (▷) [F1] (CLR) [3] (ALLEQU) deletes Eqn memory, Ans memory, and Replay memory contents.

You can input up to 255 bytes of data into the input area.

SET UP Items

- **Angle** ... Unit of angular measurement specification
  - {**Deg**}/{**Rad**} ... {degrees}/{radians}
- **Answer Type** ... Result range specification
  - {**Real**}/{**Cplx**} ... {real number}/{complex number}
- **Display** ... Display format specification (for approx only)
  - {**Fix**}/{**Sci**}/{**Norm**} ... {number of decimal places}/{number of significant digits}/{normal display format}

■ Graph Function

Pressing [F5](GRPH) displays the graph formula screen, which you can use to input a graph formula. Press [F4](G•VAR) if you want to input a graph memory.
You can also use the [F1](SEL), [F2](DEL), and [F3](TYPE) functions while the graph formula screen is on the display.

Press [F6](DRAW) to draw a graph.

■ RECALL ANS Function

Pressing [F6](▷)[F3](R•ANS) recalls Ans Memory contents.

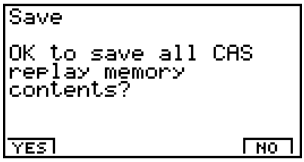
■ Solution Memory

In the CAS Mode or ALGEBRA Mode, you can save the history of a calculation you perform (replay memory contents) into solution memory. This section describes how you can access and work with the contents of solution memory. Pressing [F6](▷)[F4](MEM) on the CAS Mode or ALGEBRA Mode main menu display the initial solution memory screen shown below.

- {**SAVE**} ... {saves the calculation history to solution memory}
- {**DEL•A**}... {deletes solution memory contents}
- {**OPT**} ... {optimizes solution memory}
- {**DISP**} ... {displays solution memory contents}

• To save a calculation history to solution memory (Save)

On the initial solution memory screen, press [F1] (SAVE).



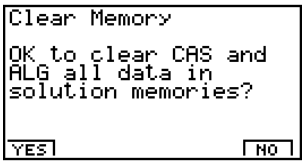
Press [F1] (YES) to save the calculation history to solution memory.

Pressing [ESC] returns to the solution memory initial screen.

- Pressing [F6] (NO) in place of [F1] (YES) returns to the solution memory initial screen without saving anything.

• To clear solution memory contents (Clear Memory)

On the initial solution memory screen, press [F2] (DEL・A).



Press [F1] (YES) to clear solution memory contents.

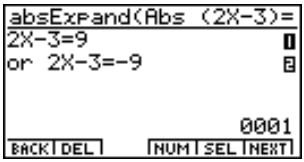
Pressing [ESC] returns to the solution memory initial screen.

- Pressing [F6] (NO) in place of [F1] (YES) returns to the solution memory initial screen without clearing anything.
- This clears both CAS Mode and ALGEBRA Mode memory contents. You cannot select the mode shows memory contents you want to delete.

• To display solution memory contents (Display Memory)

On the initial solution memory screen, press [F6] (DISP).

This displays the oldest expression and result in solution memory. The bottom line shows the record number.



- [F6] (DISP) is disabled when there is no data in Solution memory.

• To display the next record

Press [F6] (NEXT).

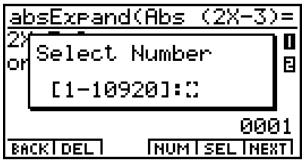
• To display the previous record

Press [F1] (BACK).

- Pressing [F1] (BACK) while the oldest record is on the display returns to the solution memory initial screen.

• To display a particular record

Press [F5] (SEL) and then input the number of the record you want to display.



Pressing [EXE] displays the record whose number you input.

• To delete a single solution memory record

Display the record you want to delete, and then press [F2] (DEL).

In response to the confirmation message that appears, press [EXE] (Yes) to delete the record you displayed.

To clear the above screen without deleting anything, press [ESC] (No).

• To toggle record number display on and off

Press [F4] (NUM) to toggle display of the record number on and off.

• To optimize solution memory (Optimization)

On the initial solution memory screen, press [F3] (OPT).

Pressing [ESC] returns to the solution memory initial screen.

Optimizing solution memory rearranges data and can free up more storage space. Perform the above procedure when solution memory capacity starts running low.

Algebra Command Reference

The following are the abbreviations used in this section.

- **Exp** ... Expression (value, formula, variable, etc.)
- **Eq** ... Equation
- **Ineq** ... Inequality
- **List** ... List
- **Mat** ... Matrix
- **Vect** ... Vector

Anything enclosed within square brackets can be omitted.

• expand

Function: Expands an expression.

Syntax: expand ({Exp/Eq/Ineq/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To expand $(X+2)^2$**

F1 (TRNS) 1 (expand) (X,θ,T + 2) x^2 EXE $X^2 + 4X + 4$

• rFactor (rFctor)

Function: Factors an expression up to its root.

Syntax: rFactor ({Exp/Eq/Ineq/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To factor the $X^2 - 3$**

F1 (TRNS) 2 (rFctor) X,θ,T x^2 − 3 EXE $(X - \sqrt{3})(X + \sqrt{3})$

• factor

Function: Factors an expression.

Syntax: factor ({Exp/Eq/Ineq/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To factor $X^2 - 4X + 4$**

F1 (TRNS) 3 (factor) X,θ,T x^2 − 4 X,θ,T + 4 EXE $(X - 2)^2$

• solve

Function: Solves an equation.

Syntax: solve(Eq [,variable] [ )]

solve({Eq-1,..., Eq-*n*}, {variable-1,...,variable-*n*} [ )]

● ● ● ● ●

Example **To solve AX + B = 0 for X**

[F1](TRNS)[4](solve)[ALPHA][X,θ,T](A)[X,θ,T][+]
[ALPHA][log](B)[SHIFT][•](=)[0][EXE]

$X = \frac{-B}{A}$

● ● ● ● ●

Example **To solve simultaneous linear equation 3X + 4Y = 5, 2X – 3Y = – 8**

[F1](TRNS)[4](solve)[SHIFT][×]({)
[3][ALPHA][+](X)[+][4][ALPHA][−](Y)[SHIFT][•](=)[5][,]
[2][ALPHA][+](X)[−][3][ALPHA][−](Y)[SHIFT][•](=)[(−)][8]
[SHIFT][÷](})[,][SHIFT][×]({)[ALPHA][+](X)[,] **X = – 1**
[ALPHA][−](Y)[SHIFT][÷](})[EXE] **Y = 2**

• X is the default when no variable is specified.

• tExpand (tExpnd)

Function: Employs the addition theorem to expand a trigonometric function.

Syntax: tExpand({Exp/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To employ the addition theorem to expand sin(A+B)**

[F1](TRNS)[5](TRIG)[1](tExpnd)
[sin][(][ALPHA][X,θ,T](A)[+][ALPHA][log](B)[EXE] **cos(B) • sin(A) + sin(B) • cos(A)**

• tCollect (tCollc)

Function: Employs the addition theorem to transform the product of a trigonometric function to a sum.

Syntax: tCollect({Exp/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To employ the addition theorem to transform sin(A)cos(B) to trigonometric sum**

[F1](TRNS)[5](TRIG)[2](tCollc)
[sin][ALPHA][X,θ,T](A)[cos][ALPHA][log](B)[EXE]

$\frac{\sin(A+B)}{2} + \frac{\sin(A-B)}{2}$

• trigToExp (trigToE)

Function: Transforms a trigonometric or hyperbolic function to an exponential function.

Syntax: trigToExp({Exp/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To convert cos(iX) to an exponential function**

F1 (TRNS) 5 (TRIG) 3 (trigToE) cos SHIFT 0 (*i*) X,θ,T EXE $\dfrac{e^{X}+e^{-X}}{2}$

• expToTrig (expToT)

Function: Converts an exponential function to a trigonometric or hyperbolic function.

Syntax: expToTrig({Exp/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To convert e^{iX} to a trigonometric function**

F1 (TRNS) 5 (TRIG) 4 (expToT)

SHIFT ln (e^{x}) (SHIFT 0 (*i*) X,θ,T EXE **cos(X) + sin(X) • i**

• simplify (smplfy)

Function: Simplifies an expression.

Syntax: simplify({Exp/Eq/Ineq/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To simplify 2X + 3Y – X + 3 = Y + X – 3Y + 3 – X**

F1 (TRNS) 6 (smplfy) 2 ALPHA + (X) + 3 ALPHA − (Y)
− ALPHA + (X) + 3 SHIFT • (=) ALPHA − (Y)
+ ALPHA + (X) − 3 ALPHA − (Y) + 3 −
ALPHA + (X) EXE **X + 3Y + 3 = –2Y + 3**

• combine (combin)

Function: Adds and reduces rational expressions.

Syntax: combine({Exp/Eq/Ineq/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To reduce the fraction (X + 1)/(X + 2) + X (X + 3)**

F1 (TRNS) 7 (combin) (X,θ,T + 1) ÷
(X,θ,T + 2) + X,θ,T (X,θ,T + 3) EXE

$$\frac{X^3 + 5X^2 + 7X + 1}{X + 2}$$

• collect (collct)

Function: Rearranges an expression, focusing on a particular variable.

Syntax: collect({Exp/Eq/Ineq/List/Mat/Vect} [,{Exp/variable}] [ )]

● ● ● ● ●

Example **To rearrange $X^2 + AX + BX$, focusing on the variable X**

F1 (TRNS) 8 (collct) X,θ,T x^2 + ALPHA X,θ,T (A) X,θ,T +
ALPHA log (B) X,θ,T EXE

$X^2 + (A + B)X$

• X is the default when nothing is specified for [,{Exp/variable}].

• substitute (sbstit)

Function: Assigns an expression to a variable.

Syntax: substitute({Exp/Eq/Ineq/List/Mat/Vect}, variable=expression
[,..., variable=expression] [ )]

● ● ● ● ●

Example **To assign 5 to X in 2X – 1**

F1 (TRNS) 9 (sbstit) 2 X,θ,T − 1 ,
X,θ,T SHIFT • (=) 5 EXE

9

• cExpand (cExpnd)

Function: Expands xth root of imaginary number.

Syntax: cExpand({Exp/Eq/Ineq/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To expand $\sqrt{2}i$**

F1 (TRNS) X,θ,T (cExpnd) SHIFT x^2 ($\sqrt{\ }$) 2 SHIFT 0 (i) EXE **1 + i**

• approx

Function: Produces a numerical approximation for an expression.

Syntax: approx({Exp/Eq/Ineq/List/Mat/Vect} [ )]

● ● ● ● ●

Example **To obtain a numerical value for $\sqrt{2}$**

F1 (TRNS) log (approx) SHIFT x^2 ($\sqrt{\ }$) 2 EXE **1.414213562**

● ● ● ● ●

Example 9^{20}

Normal: 9 ^ 2 0 EXE **12157665459056928801**

approx: F1 (TRNS) log (approx) 9 ^ 2 0 EXE **1. 215766546E+19 (Display: Norm1)**

 # About approx

With normal calculations (when approx is not used) in the CAS Mode, calculation results are displayed in full, without using exponents. When you use approx in the CAS Mode, however, results are displayed using the exponential format range specified by the Display item of the SET UP screen.

This means approx displays results in the CAS Mode the same way they are displayed in the RUN•MAT Mode.

• diff

Function: Differentiates an expression.

Syntax: diff({Exp/List} [, variable, order, derivative] [ )]
diff({Exp/List}, variable [, order, derivative] [ )]
diff({Exp/List}, variable, order [, derivative] [ )]

● ● ● ● ●

Example **To differentiate X^6 with respect to X**

F2 (CALC) 1 (diff) X,θ,T ^ 6 EXE $6X^5$

• X is the default when no variable is specified.
• 1 is the default when no order is specified.

• ∫

Function: Integrates an expression.

Syntax: ∫({Exp/List} [, variable, integration constant] [ )]
∫({Exp/List}, variable [, integration constant] [ )]
∫({Exp/List}, variable, lower limit, upper limit [ )]

● ● ● ● ●

Example **To integrate X^2 with respect to X**

F2 (CALC) 2 (∫) X,θ,T x^2 EXE $\frac{X^3}{3}$

• X is the default when no variable is specified.

• lim

Function: Determines the limits of a function expression.

Syntax: lim({Exp/List}, variable, point [, direction] [ )]

● ● ● ● ●

Example **To determine the limits of sin(X)/X when X = 0**

F2 (CALC) 3 (lim) sin X,θ,T ÷ X,θ,T , X,θ,T , 0 EXE 1

• Direction can be positive (from right) or negative (from left).

• Σ

Function: Calculates a sum.

Syntax: Σ({Exp/List}, variable, start value, end value [ )]

● ● ● ● ●

Example **To calculate the sum as the value of X in X^2 changes from X = 1 through X = 10**

F2 (CALC) 4 (Σ) X,θ,T x^2 , X,θ,T , 1 , 1 0 EXE **385**

• Π

Function: Calculates a product.

Syntax: Π({Exp/List}, variable, start value, end value [ )]

● ● ● ● ●

Example **To calculate the product as the value of X in X^2 changes from X = 1 through X = 5**

F2 (CALC) 5 (Π) X,θ,T x^2 , X,θ,T , 1 , 5 EXE **14400**

• taylor

Function: Finds a Taylor polynomial.

Syntax: taylor({Exp/List}, variable, order [, center point] [ )]

● ● ● ● ●

Example **To find a 5th order Taylor polynomial for sin(X) with respect to X = 0**

F2 (CALC) 6 (taylor) sin X,θ,T , X,θ,T , 5 , 0 EXE $\frac{X^5}{120} - \frac{X^3}{6} + X$

• The default center point is zero.

• arcLen

Function: Returns the arc length.

Syntax: arcLen({Exp/List}, variable, start value, end value [ )]

● ● ● ● ●

Example **To determine the arc length for X^2 from X = 0 to X = 1**

F2 (CALC) 7 (arcLen)
X,θ,T x^2 , X,θ,T , 0 , 1 EXE $\frac{\ln(4\sqrt{5}+8)}{4} - \frac{\ln(2)}{2} + \frac{\sqrt{5}}{2}$

• tanLine (tanLin)

Function: Returns the expression for a tangent line.

Syntax: tanLine({Exp/List}, variable, variable value at point of tangency [ )]

● ● ● ● ●

Example **To determine the expression for a line tangent with X^3 when X = 2**

F2 (CALC) 8 (tanLin) X,θ,T ^ 3 , X,θ,T , 2 EXE **12X – 16**

• denominator (den)

Function: Extracts the denominator of a fraction.

Syntax: denominator({Exp/List} [ )]

● ● ● ● ●

Example **To extract the denominator of the fraction (X + 2)/(Y – 1)**

F2 (CALC) 9 (EXTRCT) 1 (den)

(ALPHA + (X) + 2) ÷ (ALPHA − (Y) − 1 EXE **Y – 1**

• numerator (num)

Function: Extracts the numerator of a fraction.

Syntax: numerator({Exp/List} [ )]

● ● ● ● ●

Example **To extract the numerator of the fraction (X + 2)/(Y – 1)**

F2 (CALC) 9 (EXTRCT) 2 (num)

(ALPHA + (X) + 2) ÷ (ALPHA − (Y) − 1 EXE **X + 2**

• gcd

Function: Returns the greatest common divisor.

Syntax: gcd({Exp/List}, {Exp/List} [ )]

● ● ● ● ●

Example **To determine the greatest common divisor of X + 1 and X^2 – 3X – 4**

F2 (CALC) X,θ,T (gcd) X,θ,T + 1 , X,θ,T x^2 −

3 X,θ,T − 4 EXE **X + 1**

• lcm

Function: Obtains the least common multiple of two expressions

Syntax: lcm({Exp/List}, {Exp/List} [ )]

● ● ● ● ●

Example **To obtain the least common multiple of $X^2 - 1$ and $X^2 + 2X - 3$**

F2 (CALC) log (lcm) X,θ,T x^2 − 1 ,

X,θ,T x^2 + 2 X,θ,T − 3 EXE **$X^3 + 3X^2 - X - 3$**

• rclEqn

Function: Recalls multiple eqn memory contents.

Syntax: rclEqn(memory number [, ..., memory number] [ )]

● ● ● ● ●

Example **To recall the contents of equation memory 2 and equation memory 3**

F3 (EQUA) 2 (rclEqn) 2 , 3 EXE **3X – Y = 7**

3X + 6Y = 63

• The memory numbers of equations produced as the result of a recall are not updated.

• rclAllEqn (rclAll)

Function: Recall all eqn memory contents.

Syntax: rclAllEqn

• The memory numbers of equations produced as the result of a recall are not updated.

• rewrite (rewrit)

Function: Moves the right side expression to the left side.

Syntax: rewrite({Eq/Ineq/List} [ )]

● ● ● ● ●

Example **To move the right side expression of $X + 3 = 5X - X^2$ to the left side**

F3 (EQUA) 4 (rewrit) X,θ,T + 3 SHIFT • (=)

5 X,θ,T − X,θ,T x^2 EXE **$X^2 - 4X + 3 = 0$**

• exchange (exchng)

Function: Exchanges the right-side and left-side expressions.

Syntax: exchange({Eq/Ineq/List} [ )]

● ● ● ● ●

Example **To exchange the left-side and right-side expressions of 3 > 5X – 2Y**

F3 (EQUA) 5 (exchng) 3 F3 (EQUA) 1 (INEQUA) 1 (>)
5 ALPHA + (X) − 2 ALPHA − (Y) EXE **5X – 2Y < 3**

• eliminate (elim)

Function: Assigns an expression to a variable.

Syntax: eliminate({Eq/Ineq/List} -1, variable, Eq-2 [ )]

● ● ● ● ●

Example **To transform Y = 2X + 3 to X= and then substitute into 2X + 3Y = 5**

F3 (EQUA) 6 (elim) 2 ALPHA + (X) + 3 ALPHA − (Y) SHIFT • (=)
5 , ALPHA + (X) , ALPHA − (Y) SHIFT • (=)
2 ALPHA + (X) + 3 EXE **4Y – 3 = 5**

• getRight (getRgt)

Function: Gets the right-side element.

Syntax: getRight({Eq/Ineq/List} [ )]

● ● ● ● ●

Example **To extract the right side element of $Y = 2X^2 + 3X + 5$**

F3 (EQUA) 7 (getRgt) ALPHA − (Y) SHIFT • (=)
2 ALPHA + (X) x^2 + 3 ALPHA + (X) + 5 EXE **$2X^2 + 3X + 5$**

• invert

Function: Inverts two variables.

Syntax: invert({Exp/Eq/Ineq/List} [,variable name 1, variable name 2] [ )]

If you omit the variable names, variables X and Y are inverted.

● ● ● ● ●

Example **To invert X and Y in the expression 2X = Y**

F3 (EQUA) 8 (invert) 2 X,θ,T SHIFT • (=) ALPHA − (Y) EXE **2Y = X**

- **absExpand (absExp)**

  Function: Divides an expression that contains an absolute value into two expressions.

  Syntax: absExpand({Eq/Ineq} [ )]

  ● ● ● ● ●

  Example **To strip the absolute value from | 2X – 3 | = 9**

  [F3](EQUA) [9](absExp) [OPTN] [F5](Abs) [(] **2X – 3 = 9** **1**

  [2] [X,θ,T] [−] [3] [)] [SHIFT] [•](=) [9] [EXE] **or 2X – 3 = – 9** **2**

- **andConnect (andCon)**

  Function: Connects two inequalities into a single expression.

  Syntax: andConnect(Ineq-1, Ineq-2 [ )]

  ● ● ● ● ●

  Example **To combine X > – 1 and X < 3 into a single inequality**

  [F3](EQUA) [X,θ,T](andCon) [X,θ,T] [F3](EQUA) [1](INEQUA) [1](>)

  [(−)] [1] [,] [X,θ,T] [F3](EQUA) [1](INEQUA) [2](<) [3] [EXE] **–1 < X < 3**

- **eqn**

  Function: Recalls eqn memory contents.

  Syntax: eqn(memory number [ )]

  ● ● ● ● ●

  Example **To add 15 to both sides of the equation 6X – 15 = X – 7, which is stored in equation memory 3**

  [F4](eqn) [3] [)] [+] [1] [5] [EXE] **6X = X + 8**

• clear (clrVar)

Function: Clears the contents of specific equation (A to Z, r, θ).*1

Syntax: clear(variable [ )]
clear({variable list} [ )]

● ● ● ● ●

Example **To clear the contents of variable A**

F6 (▷) F1 (CLR) 1 (clrVar) ALPHA X,θ,T (A) EXE **{ }**

● ● ● ● ●

Example **To clear the contents of variables X, Y, and Z**

F6 (▷) F1 (CLR) 1 (clrVar) SHIFT × ({) ALPHA + (X) , ALPHA − (Y) , ALPHA 0 (Z) SHIFT ÷ (}) EXE **{ }**

• clearVarAll (VarAll)

Function: Clears the contents of all 28 variables (A to Z, r, θ).

Syntax: clearVarAll **{ }**

*1 When you start out with memories A, B, C, and D, for example, and delete memories A and B, the display shows only C,D because they are the only memories remaining.

■ List Calculation Commands [OPTN]-[LIST]

• Dim

Function: Returns the dimension of a list.

Syntax: Dim List

● ● ● ● ●

Example **To determine the dimension of list {1, 2, 3}**

OPTN F1(LIST) 1(CALC) 1(Dim) SHIFT ×({) 1 , 2 , 3 SHIFT ÷(}) EXE **3**

• Min

Function: Returns the minimum value of an expression or the elements in a list.

Syntax: Min({List/Exp} [ )]

Min({List/Exp}, {List/Exp} [ )]

● ● ● ● ●

Example **To determine the minimum value of the elements in list {1, 2, 3}**

OPTN F1(LIST) 1(CALC) 2(Min) SHIFT ×({) 1 , 2 , 3 SHIFT ÷(}) EXE **1**

● ● ● ● ●

Example **To compare each element of list {1, 2, 3} with the value 2, and produce a list whose elements are the minimum value resulting from each comparison**

OPTN F1(LIST) 1(CALC) 2(Min) SHIFT ×({) 1 , 2 , 3 SHIFT ÷(}) , 2 EXE **{1, 2, 2}**

● ● ● ● ●

Example **To compare the elements of list {1, 2, 3} and list {3, 1, 2}, and produce a list whose elements are the minimum value resulting from each comparison**

OPTN F1(LIST) 1(CALC) 2(Min) SHIFT ×({) 1 , 2 , 3 SHIFT ÷(}) , SHIFT ×({) 3 , 1 , 2 SHIFT ÷(}) EXE **{1, 1, 2}**

• Max

Function: Returns the maximum value of an expression or the elements of a list.

Syntax: Max({List/Exp} [ )]

Max({List/Exp}, {List/Exp} [ )]

● ● ● ● ●

Example **To determine the maximum value of the elements in list {1, 2, 3}**

OPTN F1 (LIST) 1 (CALC) 3 (Max) SHIFT × ({) 1 , 2 , 3 SHIFT ÷ (}) EXE **3**

● ● ● ● ●

Example **To compare each element of list {1, 2, 3} with the value 2, and produce a list whose elements are the maximum value resulting from each comparison**

OPTN F1 (LIST) 1 (CALC) 3 (Max) SHIFT × ({) 1 , 2 , 3 SHIFT ÷ (}) , 2 EXE **{2, 2, 3}**

● ● ● ● ●

Example **To compare the elements of list {1, 2, 3} and list {3, 1, 2}, and produce a list whose elements are the maximum value resulting from each comparison**

OPTN F1 (LIST) 1 (CALC) 3 (Max) SHIFT × ({) 1 , 2 , 3 SHIFT ÷ (}) , SHIFT × ({) 3 , 1 , 2 SHIFT ÷ (}) EXE **{3, 2, 3}**

• Mean

Function: Returns the mean of the elements in a list.

Syntax: Mean(List [ )]

Mean(List, List [ )]

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the mean of the elements in list {1, 2, 3}**

OPTN F1 (LIST) 1 (CALC) 4 (Mean) SHIFT × ({) 1 , 2 , 3 SHIFT ÷ (}) EXE **2**

● ● ● ● ●

Example **To determine the mean of the elements in list {1, 2, 3} when their frequencies are {3, 2, 1}**

OPTN F1 (LIST) 1 (CALC) 4 (Mean) SHIFT × ({) 1 , 2 , 3
SHIFT ÷ (}) , SHIFT × ({) 3 , 2 , 1 SHIFT ÷ (}) EXE

$\frac{5}{3}$

• Median

Function: Returns the median of the elements in a list.

Syntax: Median(List [ )]

Median(List, List [ )]

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the median of the elements in list {1, 2, 3}**

OPTN F1 (LIST) 1 (CALC) 5 (Median) SHIFT × ({) 1 , 2 , 3
SHIFT ÷ (}) EXE

2

● ● ● ● ●

Example **To determine the median of the elements in list {1, 2, 3} when their frequencies are {3, 2, 1}**

OPTN F1 (LIST) 1 (CALC) 5 (Median) SHIFT × ({) 1 , 2 , 3
SHIFT ÷ (}) , SHIFT × ({) 3 , 2 , 1 SHIFT ÷ (}) EXE

$\frac{3}{2}$

• Sum

Function: Returns the sum of the elements in a list.

Syntax: Sum List

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the sum of the elements in list {1, 2, 3}**

OPTN F1 (LIST) 1 (CALC) 6 (Sum) SHIFT × ({) 1 , 2 , 3
SHIFT ÷ (}) EXE

6

• Prod

Function: Returns the product of the elements in a list.

Syntax: Prod List

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the product of the elements in list {2, 3, 4}**

OPTN F1 (LIST) 1 (CALC) 7 (Prod) SHIFT × ({) 2 , 3 , 4 SHIFT ÷ (}) EXE **24**

• Cuml

Function: Returns the cumulative frequency of the elements in a list.

Syntax: Cuml List

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the cumulative frequency of the elements in list {1, 2, 3}**

OPTN F1 (LIST) 1 (CALC) 8 (Cuml) SHIFT × ({) 1 , 2 , 3 SHIFT ÷ (}) EXE **{1, 3, 6}**

• Percent (%)

Function: Returns the percentage of each element in a list, the sum of which is assumed to be 100.

Syntax: Percent List

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the percentage of each element in the list {1, 2, 3}**

OPTN F1 (LIST) 1 (CALC) 9 (%) SHIFT × ({) 1 , 2 , 3 SHIFT ÷ (}) EXE $\left\{\frac{50}{3}, \frac{100}{3}, 50\right\}$

• ⊿List

Function: Returns a list whose elements are the differences between the elements of another list.

Syntax: ⊿ List List

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To generate a list whose elements are the differences between the elements of list {1, 2, 4}**

OPTN F1 (LIST) 1 (CALC) X,θ,T (⊿List) SHIFT × ({) 1 , 2 , 4 SHIFT ÷ (}) EXE

{1, 2}

• StdDev

Function: Returns the sample standard deviation of the elements in a list.

Syntax: StdDev List

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the sample standard deviation of the elements in list {1, 2, 4}**

OPTN F1 (LIST) 1 (CALC) log (StdDev) SHIFT × ({) 1 , 2 , 4 SHIFT ÷ (}) EXE

$\frac{\sqrt{21}}{3}$

• Variance (Vari)

Function: Returns the variance of the elements in a list.

Syntax: Variance List

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To determine the variance of the elements in list {1, 2, 4}**

OPTN F1 (LIST) 1 (CALC) ln (Vari) SHIFT × ({) 1 , 2 , 4 SHIFT ÷ (}) EXE

$\frac{7}{3}$

• Seq

Function: Generates a list in accordance with a numeric sequence expression.

Syntax: Seq(Exp, variable, start value, end value, [increment] [ )]

If you do not specify an increment, an increment of 1 is used.

● ● ● ● ●

Example **To generate a list in accordance with the expression: value A, end value 3A, increment A**

OPTN F1 (LIST) 2 (CREATE) 1 (Seq) X,θ,T ▸ X,θ,T ▸ ALPHA X,θ,T (A) ▸ 3 ALPHA X,θ,T (A) ▸ ALPHA X,θ,T (A) EXE **{A, 2A, 3A}**

• Augment (Augmnt)

Function: Returns a new list that appends List 2 to List 1.

Syntax: Augment(List, List [ )]

● ● ● ● ●

Example **To combine list {1, 2} and list {3, 4}**

OPTN F1 (LIST) 2 (CREATE) 2 (Augmnt) SHIFT × ({) 1 ▸ 2 SHIFT ÷ (}) ▸ SHIFT × ({) 3 ▸ 4 SHIFT ÷ (}) EXE **{1, 2, 3, 4}**

• Fill

Function: Replaces the elements of a list with a specified value or expression. This command can also be used to create a new list whose elements all contain the same value or expression.

Syntax: Fill({Exp/Eq/Ineq}, List [ )]

Fill(Exp, numeric value [ )]

● ● ● ● ●

Example **To replace the elements of list {3, 4} with X**

OPTN F1 (LIST) 2 (CREATE) 3 (Fill) X,θ,T ▸ SHIFT × ({) 3 ▸ 4 SHIFT ÷ (}) EXE **{X, X}**

● ● ● ● ●

Example **To create a list with eight elements, all of which are X**

OPTN F1 (LIST) 2 (CREATE) 3 (Fill) X,θ,T ▸ 8 EXE **{X, X, X, X, X, X, X, X}**

• SortA

Function: Sorts the elements of a list into ascending order.

Syntax: SortA(List [ )]

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To sort the elements of list {1, 5, 3} into ascending order**

OPTN F1 (LIST) 2 (CREATE) 4 (SortA) SHIFT × ({) 1 , 5 , 3 SHIFT ÷ (}) EXE **{1, 3, 5}**

• SortD

Function: Sorts the elements of a list into descending order.

Syntax: SortD(List [ )]

The list must contain values or mathematical expressions only. Equations and inequalities are not allowed.

● ● ● ● ●

Example **To sort the elements of list {1, 5, 3} into descending order**

OPTN F1 (LIST) 2 (CREATE) 5 (SortD) SHIFT × ({) 1 , 5 , 3 SHIFT ÷ (}) EXE **{5, 3, 1}**

• SubList (SubLst)

Function: Extracts a specific section of a list into a new list.

Syntax: SubList(List, start number [, end number] [ )]

● ● ● ● ●

Example **To extract element 2 through element 3 from list {1, 2, 3, 4}**

OPTN F1 (LIST) 2 (CREATE) 6 (SubLst) SHIFT × ({) 1 , 2 , 3 , 4 SHIFT ÷ (}) , 2 , 3 EXE **{2, 3}**

- If you do not specify an end number, all the elements from the start number to the end of the list are extracted.

• List→Mat (L→Mat)

Function: Converts lists into a matrix.

Syntax: List→Mat(List [, ... ,List] [ )]

● ● ● ● ●

Example **To convert list {3, 5} and list {2, 4} into a matrix**

[OPTN] [F1] (LIST) [3] (LIST→) [1] (L→Mat) [SHIFT] [×] ({) [3] [,] [5]
[SHIFT] [÷] (}) [,] [SHIFT] [×] ({) [2] [,] [4] [SHIFT] [÷] (}) [EXE]

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

• List→Vect (L→Vect)

Function: Converts a list into a vector.

Syntax: List→Vect List

● ● ● ● ●

Example **To convert list {3, 2} into a vector**

[OPTN] [F1] (LIST) [3] (LIST→) [2] (L→Vect) [SHIFT] [×] ({) [3] [,] [2]
[SHIFT] [÷] (}) [EXE]

[3, 2]

■ Matrix Calculation Commands [OPTN]-[MAT]

• Dim

Function: Returns the dimensions of a matrix.

Syntax: Dim Mat

● ● ● ● ●

Example **To determine the dimensions of the matrix below**

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

OPTN F2 (MAT) 1 (CALC) 1 (Dim) SHIFT + ([ ) SHIFT + ([ )
1 , 2 , 3 SHIFT − ( ]) SHIFT + ([ ) 4 , 5 , 6
SHIFT − ( ]) SHIFT − ( ]) EXE **{2, 3}**

• Det

Function: Returns the determinant of a matrix.

Syntax: Det Mat

● ● ● ● ●

Example **To determine the determinant of the matrix below**

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

OPTN F2 (MAT) 1 (CALC) 2 (Det) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 4 , 5
SHIFT − ( ]) SHIFT − ( ]) EXE **– 3**

• Norm

Function: Returns the norm of a matrix.

Syntax: Norm Mat

● ● ● ● ●

Example **To determine the norm of the matrix below**

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

OPTN F2 (MAT) 1 (CALC) 3 (Norm) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 4 , 5
SHIFT − ( ]) SHIFT − ( ]) EXE **$\sqrt{46}$**

• EigVc

Function: Returns the eigenvector of a matrix.

Syntax: EigVc Mat

● ● ● ● ●

Example **To determine the eigenvector of the matrix below**

$$\begin{bmatrix} 3 & 4 \\ 1 & 3 \end{bmatrix}$$

OPTN F2 (MAT) 1 (CALC) 4 (EigVc)
SHIFT + ([ ) SHIFT + ([ ) 3 , 4
SHIFT − ( ]) SHIFT + ([ ) **[0.894427191 – 0.894427191]**
1 , 3 SHIFT − ( ]) SHIFT − ( ]) EXE **[0.4472135955 0.4472135955]**

Eigenvectors are stacked vertically on the display.

In this example, (0.894427191 0.4472135955) are the eigenvectors that correspond to 5, while (–0.894427191 0.4472135955) are the eigenvectors that correspond to 1.

An eigenvector has an infinite number of solutions. The eigenvector displayed by this command is the one with a size of 1.

• EigVl

Function: Returns the eigenvalue of a matrix.

Syntax: EigVl Mat

● ● ● ● ●

Example **To determine the eigenvalue of the matrix below**

$$\begin{bmatrix} 3 & 4 \\ 1 & 3 \end{bmatrix}$$

OPTN F2 (MAT) 1 (CALC) 5 (EigVl) SHIFT + ([ ) SHIFT + ([ )
3 , 4 SHIFT − ( ]) SHIFT + ([ ) 1 , 3
SHIFT − ( ]) SHIFT − ( ]) EXE **{5, 1}**

• Rref

Function: Returns the reduced row echelon form of a matrix.

Syntax: Rref Mat

● ● ● ● ●

Example **To determine the reduced row echelon form of the matrix below**

$$\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}$$

OPTN F2 (MAT) 1 (CALC) 6 (Rref) SHIFT + ([ ) SHIFT + ([ )
(−) 2 , (−) 2 , 0 , (−) 6 SHIFT − ( ]) SHIFT + ([ )
1 , (−) 1 , 9 , (−) 9 SHIFT − ( ])
SHIFT + ([ ) (−) 5 , 2 , 4 , (−) 4
SHIFT − ( ]) SHIFT − ( ]) EXE

$$\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & -\frac{62}{71} \end{bmatrix}$$

• Ref

Function: Returns the row echelon form of a matrix.

Syntax: Ref Mat

● ● ● ● ●

Example **To determine the row echelon form of the matrix below**

$$\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}$$

OPTN F2 (MAT) 1 (CALC) 7 (Ref) SHIFT + ([ ) SHIFT + ([ )
(−) 2 , (−) 2 , 0 , (−) 6 SHIFT − ( ]) SHIFT + ([ )
1 , (−) 1 , 9 , (−) 9 SHIFT − ( ])
SHIFT + ([ ) (−) 5 , 2 , 4 , (−) 4
SHIFT − ( ]) SHIFT − ( ]) EXE

$$\begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & 1 & -\frac{9}{2} & 6 \\ 0 & 0 & 1 & -\frac{62}{71} \end{bmatrix}$$

• LU

Function: Returns the LU resolution of a matrix.

Syntax: LU(Mat, lower memory, upper memory)

● ● ● ● ●

Example **To determine the LU resolution of the matrix below**

$$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$$

The lower matrix is assigned to variable A, while the upper matrix is assigned to variable B.

OPTN F2 (MAT) 1 (CALC) 8 (LU) SHIFT + ([ ) SHIFT + ([ )
6 , 1 2 , 1 8 SHIFT − ( ]) SHIFT + ([ )
5 , 1 4 , 3 1 SHIFT − ( ]) SHIFT + ([ )
3 , 8 , 1 8 SHIFT − ( ]) SHIFT − ( ]) ▶
ALPHA X,θ,T (A) ▶ ALPHA log (B) EXE

$$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$$

The upper matrix is displayed as the calculation result.

To display the lower matrix, recall the lower matrix variable (A in this example) specified by the command.

ALPHA X,θ,T (A) EXE

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$$

To display the upper matrix, recall the upper matrix variable (B in this example) specified by the command.

• Trn

Function: Transposes a matrix.

Syntax: Trn Mat

● ● ● ● ●

Example **To transpose the matrix below**

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

OPTN F2 (MAT) 2 (CREATE) 1 (Trn) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 3 , 4
SHIFT − ( ]) SHIFT − ( ]) EXE

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

• Augment (Augmnt)

Function: Combines two matrices.

Syntax: Augment(Mat, Mat [ )]

● ● ● ● ●

Example **To combine the two matrices below**

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

OPTN F2 (MAT) 2 (CREATE) 2 (Augmnt) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 3 , 4
SHIFT − ( ]) SHIFT − ( ]) , SHIFT + ([ ) SHIFT + ([ )
5 , 6 SHIFT − ( ]) SHIFT + ([ ) 7 , 8
SHIFT − ( ]) SHIFT − ( ]) EXE

$$\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

• Identify (Ident)

Function: Creates an identity matrix

Syntax: Ident numeric value

● ● ● ● ●

Example **To create a 2 × 2 identity matrix**

OPTN F2 (MAT) 2 (CREATE) 3 (Ident) 2 EXE

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

• Fill

Function: Replaces the elements of a matrix with a specified value or expression.

This command can also be used to create a new matrix whose elements all contain the same value or expression.

Syntax: Fill(Exp, Mat [ )]

Fill(Exp, number of lines, number of rows [ )]

● ● ● ● ●

Example **To replace the elements of the matrix below with X**

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

OPTN F2 (MAT) 2 (CREATE) 4 (Fill) X,θ,T , SHIFT + ([ )
SHIFT + ([ ) 3 , 4 SHIFT − ( ]) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT − ( ]) EXE

$$\begin{bmatrix} X & X \\ X & X \end{bmatrix}$$

● ● ● ● ●

Example **To create a 2 × 3 matrix, all of whose entries are X**

OPTN F2 (MAT) 2 (CREATE) 4 (Fill) X,θ,T , 2 , 3 EXE

$$\begin{bmatrix} X & X & X \\ X & X & X \end{bmatrix}$$

• SubMat

Function: Extracts a specific section of a matrix into a new matrix.

Syntax: SubMat(Mat [, start row] [, start column] [, end row] [, end column] [ )]

● ● ● ● ●

Example **To extract the section from row 2, column 2 to row 3, column 3 from the following matrix**

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

OPTN F2 (MAT) 2 (CREATE) 5 (SubMat) SHIFT + ([ ) SHIFT + ([ )
1 , 2 , 3 SHIFT − ( ]) SHIFT + ([ ) 4 , 5 , 6
SHIFT − ( ]) SHIFT + ([ ) 7 , 8 , 9 SHIFT − ( ])
SHIFT − ( ]) , 2 , 2 , 3 , 3 EXE

$$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

- If you do not specify an end row and column, all the entries from the start row/column to the end of the matrix are extracted.

• Diag

Function: Extracts the diagonal elements of a matrix.

Syntax: Diag Mat

● ● ● ● ●

Example **To extract the diagonal elements of the matrix below**

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

OPTN F2 (MAT) 2 (CREATE) 6 (Diag) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 3 , 4
SHIFT − ( ]) SHIFT − ( ]) EXE **[1, 4]**

• Mat→List (M→List)

Function: Converts a specific column of a matrix into a list.

Syntax: Mat→List(Mat, column number [ )]

● ● ● ● ●

Example **To convert column 2 of the matrix below to a list**

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

OPTN F2 (MAT) 3 (MAT→) 1 (M→List) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 3 , 4
SHIFT − ( ]) SHIFT − ( ]) , 2 EXE **{2, 4}**

• Mat→Vect (M→Vect)

Function: Converts a specific column of a matrix into a vector.

Syntax: Mat→Vect(Mat, column number [ )]

● ● ● ● ●

Example **To convert column 2 of the matrix below to a vector**

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

OPTN F2 (MAT) 3 (MAT→) 2 (M→Vect) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 3 , 4
SHIFT − ( ]) SHIFT − ( ]) , 2 EXE **[2, 4]**

• Swap

Function: Swaps two rows of a matrix.

Syntax: Swap Mat, row number 1, row number 2

● ● ● ● ●

Example **To swap row 1 with row 2 of the following matrix**

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

OPTN F2 (MAT) 4 (ROW) 1 (Swap) SHIFT + ([ ) SHIFT + ([ )
1 , 2 SHIFT − ( ]) SHIFT + ([ ) 3 , 4
SHIFT − ( ]) SHIFT − ( ]) , 1 , 2 EXE

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

• ∗Row

Function: Returns the scalar product of a row of a matrix.

Syntax: ∗Row(Exp, Mat, row number [ )]

● ● ● ● ●

Example **To multiply row 1 of the matrix below by X**

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

OPTN F2 (MAT) 4 (ROW) 2 (∗Row) X,θ,T , SHIFT + ([ )
SHIFT + ([ ) 1 , 2 SHIFT − ( ]) SHIFT + ([ )
3 , 4 SHIFT − ( ]) SHIFT − ( ]) , 1 EXE

$$\begin{bmatrix} X & 2X \\ 3 & 4 \end{bmatrix}$$

• ∗Row+

Function: Calculates the scalar product of one row of a matrix and adds the result to another row.

Syntax: ∗Row+(Exp, Mat, line number 1, line number 2 [ )]

● ● ● ● ●

Example **To multiply row 1 of the matrix below by X, and add the result to row 2**

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

OPTN F2 (MAT) 4 (ROW) 3 (∗Row+) X,θ,T , SHIFT + ([ )
SHIFT + ([ ) 1 , 2 SHIFT − ( ]) SHIFT + ([ )
3 , 4 SHIFT − ( ]) SHIFT − ( ]) , 1 , 2 EXE

$$\begin{bmatrix} 1 & 2 \\ X+3 & 2X+4 \end{bmatrix}$$

- **Row+**

  Function: Adds one row of a matrix and to another row.

  Syntax: Row+(Mat, row number 1, row number 2 [ )]

  ● ● ● ● ●

  Example **To add row 1 of the matrix below to row 2**

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

  [OPTN] [F2] (MAT) [4] (ROW) [4] (Row+) [SHIFT] [+] ([ )
  [SHIFT] [+] ([ ) [1] [,] [2] [SHIFT] [−] ( ]) [SHIFT] [+] ([ )
  [3] [,] [4] [SHIFT] [−] ( ]) [SHIFT] [−] ( ]) [,] [1] [,] [2] [EXE]

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

■ Vector Calculation Commands [OPTN]-[VECT]

• Dim

Function: Returns the dimension of a vector.

Syntax: Dim Vect

● ● ● ● ●

Example **To determine the dimension of the vector (1 2 3)**

OPTN F3 (VECT) 1 (CALC) 1 (Dim) SHIFT + ([ ) 1 , 2 , 3 SHIFT − ( ]) EXE **3**

• CrossP

Function: Returns the cross product of two vectors.

Syntax: CrossP(Vect, Vect [ )]

● ● ● ● ●

Example **To determine the cross product of vector (1 2 3) and vector (4 5 6)**

OPTN F3 (VECT) 1 (CALC) 2 (CrossP) SHIFT + ([ ) 1 , 2 , 3 SHIFT − ( ]) , SHIFT + ([ ) 4 , 5 , 6 SHIFT − ( ]) EXE **[– 3, 6, – 3]**

• DotP

Function: Returns the dot product of two vectors.

Syntax: DotP(Vect, Vect [ )]

● ● ● ● ●

Example **To determine the dot product of vector (1 2 3) and vector (4 5 6)**

OPTN F3 (VECT) 1 (CALC) 3 (DotP) SHIFT + ([ ) 1 , 2 , 3 SHIFT − ( ]) , SHIFT + ([ ) 4 , 5 , 6 SHIFT − ( ]) EXE **32**

• Norm

Function: Returns the norm of a vector.

Syntax: Norm Vect

● ● ● ● ●

Example **To determine the norm of the vector (1 2 3)**

OPTN F3 (VECT) 1 (CALC) 4 (Norm) SHIFT + ([ ) 1 , 2 , 3 SHIFT − ( ]) EXE **$\sqrt{14}$**

• UnitV

Function: Normalizes a vector.

Syntax: UnitV Vect

● ● ● ● ●

Example **To normalize a vector (1 2 3)**

OPTN F3 (VECT) 1 (CALC) 5 (UnitV)
SHIFT + ([ ) 1 , 2 , 3
SHIFT − ( ]) EXE

$$\left[\frac{\sqrt{14}}{14}, \frac{\sqrt{14}}{7}, \frac{3\sqrt{14}}{14}\right]$$

• Angle

Function: Returns the angle formed by two vectors.

Syntax: Angle(Vect, Vect [ )]

● ● ● ● ●

Example **To determine the angle formed by vector (1 2) and vector (3 4) (Unit Angle: Rad)**

OPTN F3 (VECT) 1 (CALC) 6 (Angle) SHIFT + ([ ) 1 , 2
SHIFT − ( ]) , SHIFT + ([ ) 3 , 4 SHIFT − ( ]) EXE

$$\cos^{-1}\left(\frac{11\sqrt{5}}{25}\right)$$

• Augment (Augmnt)

Function: Combines two vectors.

Syntax: Angle(Vect, Vect [ )]

● ● ● ● ●

Example **To combine vector (1 2) and vector (3 4)**

OPTN F3 (VECT) 2 (CREATE) 1 (Augmnt) SHIFT + ([ ) 1 , 2
SHIFT − ( ]) , SHIFT + ([ ) 3 , 4 SHIFT − ( ]) EXE

[1, 2, 3, 4]

• Fill

Function: Replaces the elements of a vector with a specified value or expression.

Syntax: Fill(Exp, Vect [ )]

● ● ● ● ●

Example **To replace the components of the vector below with X**

OPTN F3 (VECT) 2 (CREATE) 2 (Fill) X,θ,T , SHIFT + ([ )
3 , 4 SHIFT − ( ]) EXE

[X, X]

• Vect→List (V→List)

Function: Converts a vector into a list.

Syntax: Vect→List Vect

● ● ● ● ●

Example **To convert vector (3 2) into a list**

[OPTN] [F3](VECT) [3](VECT→) [1](V→List) [SHIFT] [+]([ ) [3] [,] [2]
[SHIFT] [−]( ]) [EXE] **{3, 2}**

• Vect→Mat (V→Mat)

Function: Converts vectors into a matrix.

Syntax: Vect→Mat(Vect [, ... ,Vect] ( ])

● ● ● ● ●

Example **To convert vector (3 5) and (2 4) into a matrix**

[OPTN] [F3](VECT) [3](VECT→) [2](V→Mat) [SHIFT] [+]([ ) [3] [,] [5]
[SHIFT] [−]( ]) [,] [SHIFT] [+]([ ) [2] [,] [4] [SHIFT] [−]( ]) [EXE]

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

7-2 Algebra Mode

The CAS Mode automatically provides you with the final result only. The Algebra Mode, on the other hand, lets you obtain intermediate results at a number of steps along the way.

On the Main Menu, select the **ALGEBRA** icon to enter the Algebra Mode. The screens in this mode are the same as those in the CAS Mode.

Operations in the Algebra Mode are identical to those in the CAS Mode, except for a number of limitations. Also, the following commands are available in the Algebra Mode only.

• arrange (arrang)

Function: Collects like terms and arranges them in order, starting with the term that contains the smallest coefficient.

Syntax: arrange({Exp/Eq/Ineq} [ )]

● ● ● ● ●

Example **To arrange 2X + 3 – 5X + 8Y in sequence of its variables**

F1 (TRNS) 9 (arrang) 2 ALPHA + (X) + 3 −
5 ALPHA + (X) + 8 ALPHA − (Y) EXE **– 5X + 2X + 8Y + 3**

• replace (replac)

Function: Replaces a variable with the expression assigned to the corresponding expression variable.

Syntax: replace({Exp/Eq/Ineq} [ )]

● ● ● ● ●

Example **To replace S in the expression 3X + 2S, when the expression 2X + 1 is assigned to S**

F1 (TRNS) X,θ,T (replac) 3 X,θ,T + 2 ALPHA × (S) EXE **3X + 2 (2X + 1)**

7-3 Tutorial Mode

On the Main Menu, select the **TUTOR** icon to enter the Tutorial Mode.

■ Tutorial Mode Flow

1. Specify the expression type.
2. Define the expression.
3. Specify the solve mode.

■ Specifying the Expression Type

Entering the Tutorial Mode displays a menu of the following expression types.

- Linear Equation
- Linear Inequality
- Quadratic Equation
- Simul (Simultaneous) Equation

Use the cursor keys to highlight the expression type you want to specify, and then press [EXE].

This displays a list of formulas for the expression type you select. Move the cursor to the formula you want to use.

In the case of Linear Inequality, press [F4] (TYPE) to select the inequality type.

The following shows the formulas available for each type of expression.

Linear Equation — 6 Types

- $AX = B$
- $AX + B = C$
- $A(BX + C) = D(EX + F)$
- $X + A = B$
- $AX + B = CX + D$
- $|AX + B| = C$

Linear Inequality — 6 × 4 Types

- $AX \{ > < \geqq \leqq \} B$
- $AX + B \{ > < \geqq \leqq \} C$
- $A(BX + C) \{ > < \geqq \leqq \} D(EX + F)$
- $X + A \{ > < \geqq \leqq \} B$
- $AX + B \{ > < \geqq \leqq \} CX + D$
- $|AX + B| \{ > < \geqq \leqq \} C$

Quadratic Equation — 5 Types

- $AX^2 = B$
- $AX^2 + BX + C = 0$
- $AX^2 + BX + C = DX^2 + EX + F$
- $(AX + B)^2 = C$
- $AX^2 + BX + C = D$

Simul Equation — 10 Types

- $AX + BY = C$
  $DX + EY = F$
- $AX + BY + C = 0$
  $DX + EY + F = 0$
- $AX + BY = C$
  $Y = DX + E$
- $AX + BY = C$
  $DX + EY + F = GX + HY + I$
- $AX + BY + C = DX + EY + F$
  $Y = GX + H$
- $Y = AX + B$
  $Y = CX + D$
- $AX + BY + C = DX + EY + F$
  $GX + HY + I = JX + KY + L$
- $AX + BY = C$
  $DX + EY + F = 0$
- $AX + BY + C = 0$
  $Y = DX + E$
- $AX + BY + C = 0$
  $DX + EY + F = GX + HY + I$

Pressing [F6](EXCH) reverses the left side and right side elements of the expression.

■ Defining the Expression

In this step, you specify coefficients and define the expression. You can select any of the three following methods for specifying coefficients.

- {**RAND**} ... {random generation of coefficients}
- {**INPUT**} ... {key input of coefficients}
- {**SMPL**} ... {selection of coefficients from samples}
- {**SEED**} ... {selection of a number from 1 to 99 (specification of the same number displays the same expression)}

F1 (RAND) or EXE generates random coefficients and defines the expression.

F2 (INPUT) displays the coefficient input screen. Input coefficients, pressing EXE after each. After you finish inputting all the coefficients, press F6 (EXE) to define the coefficient.

F3 (SMPL) displays a number of preset sample expressions. Highlight the one you want to use and then press EXE to define it.

Pressing F4 (SEED) displays a number selection screen. When you want to create the same problem on another calculator, specify an appropriate matching number and press EXE.

No matter what method you use, the expression you define is displayed in the output area.

You can copy an expression to the Graph Mode as a graph function*1.

- {**L•COP**}/{**R•COP**} ... copy {left side element}/{right side element} as a graph function

(Simultaneous Equation Mode*2)

- {**1•COP**}/{**2•COP**} ... copy {first}/{second} expression as a graph function

*1 In the case of an inequality, the inequality symbols are also copied.

*2 Simultaneous equations are transformed to the format Y = AX + B when copied.

■ Specifying the Solve Mode

You can select one of the following three solve modes for the displayed expression.

• {**VRFY**} ... {Verify Mode}

In this mode, you input a solution for verification of whether or not it is correct. It provides a good way to check solutions you arrive at manually.

• {**MANU**} ... {Manual Mode}

In this mode, you manually input algebra commands, transform the expression, and calculate a result.

• {**AUTO**} ... {Auto Mode}

In this mode, the solution is produced automatically, one step at a time.

■ Verify Mode

Press F4(VRFY) to enter the Verify Mode.

The expression is shown in the top line of the display. Input the solution underneath it, and then press F6(JUDG) to determine whether the solution is correct.

The verification result screen shows the left side and right side verification result (except for a linear equation).

- However, in the case where a linear equation or quadratic equation has two solutions, the left side and right side are obtained for the value where the pointer is located.
- In the case of simultaneous equations where the left side and right side of the second equation are dissimilar even though the left side and right side of the first equation match, the left side and right side of the second equation only are obtained. In other cases, the left side and right side of the first equation are obtained.

The type of solution input screen that appears is selected according to the expression type. To input a different type, press F1(TYPE) and then select the solution type you want to want to use. Available solution types depend on the mode.

• {**X = a**} ... X has one solution (X = a) (linear equation default)

• {**X = a, b**} ... X has two solutions (X = a, X = b) (quadratic equation default)

• {**X = a, Y=**} ... X and Y have one solution each (X = a, Y = b) (simultaneous equation default)

• {**X > a**} ... X { > < ≧ ≦ } a (linear inequality default)

• {**X < a, b <**} ... X < a, b < X or X ≦ a, b ≦ X

• {**a < X < b**} ... a < X< b, a ≦ X ≦ b or X = a

• {**Identi**} **(Identity)** ... identity of left side and right side

• {**Many**} **(Many Solutions)** ... many solutions

• {**No sol**} **(No Solution)** ... no solution

You can press [F4] (MANU) to change to the Manual Mode or [F5] (AUTO) to change to the Auto Mode.

● ● ● ● ●

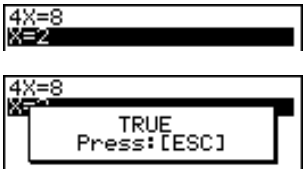
Example **To solve 4X = 8 in the Verify Mode**

(Linear Equation)(AX = B)

[F2] (INPUT) [4] [EXE] [8] [EXE] [F6] (EXE)

[F4] (VRFY) [2] [EXE]

[F6] (JUDG)



■ Manual Mode

Press [F5](MANU) to enter the Manual Mode.

As with the Algebra Mode, the screen is divided between an input area and a display area. This means you can select Algebra Mode commands from the function menu, transform the expression, and solve it.

Operation is the same as that in the Algebra Mode.

After you obtain a result, you can press [F5](JUDG) to determine whether or not it is correct.

- {**DISP**} ... Determines whether the expression in the display area is a correct solution.
- {**Identi**} ... identity of left side and right side
- {**Many**} ... many solutions
- {**No sol**} ... no solution

You can press [F6](AUTO) to change to the Auto Mode.

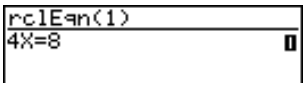
● ● ● ● ●

Example **Solve 4X = 8 in the Manual Mode**

(Linear Equation)(AX=B)

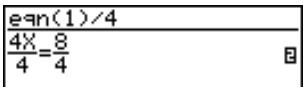
[F2](INPUT) [4] [EXE] [8] [EXE] [F6](EXE)

[F5](MANU)



[F4](eqn) [1] [)] [÷] [4]

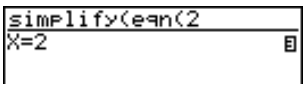
[EXE]



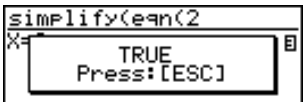
[F1](TRNS) [1](smplfy)

[F4](eqn) [2]

[EXE]



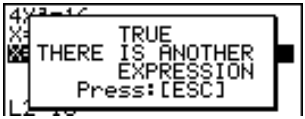
[F5](JUDG) [1](DISP)

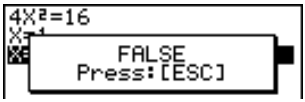


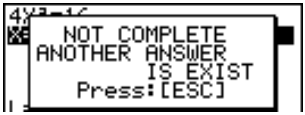
● ● ● ● ●

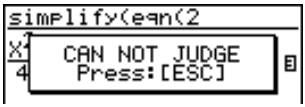
Example **$4X^2 = 16$**
True (X = 2, X = – 2)

Besides “TRUE” the messages shown below can also appear as the result of verification. “CAN NOT JUDGE” appears in the Manual Mode, while the other messages appear in both the Verify Mode and Manual Mode.









■ Auto Mode

Press [F6](AUTO) to enter the Auto Mode.

In the Simultaneous Equation Mode, you must also select SBSTIT (Substitution Method) or ADD-SU (Addition/Subtraction Method).

The Substitution Method first transforms the equation to the format Y = aX + b, and substitutes aX + b for Y*1 in the other equation.

The Addition/Subtraction Method multiplies both sides of the expression by the same value to isolate the coefficient X (or Y).

As with the Algebra Mode, the screen is divided between an input area and a display area.

Each press of [F6](NEXT) advances to the next step. [F6](NEXT) is not shown on the display when the solution is obtained.

You can scroll back through the steps by pressing [F1](BACK).

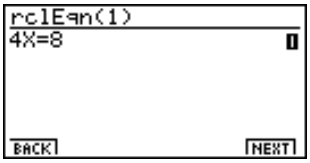
● ● ● ● ●

Example **To solve 4X = 8 in the Auto Mode**

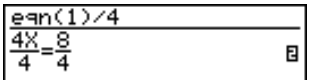
(Linear Equation)(AX = B)

[F2](INPUT) [4] [EXE] [8] [EXE] [F6](EXE)

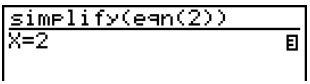
[F6](AUTO)



[F6](NEXT)



[F6](NEXT)



*1 You can press [F5](ADD SU) at any time to switch from Substitution Method to Addition / Subtraction Method.

See 7-1-8 for information about graph functions.

7-4 Algebra System Precautions

- If an algebraic operation cannot be performed for some reason, the original expression remains on the display.
- It may take considerable time to perform an algebraic operation. Failure of a result to appear immediately does not indicate malfunction of the computer.
- Any expression can be displayed in various different formats. Because of this, you should not assume that an expression is wrong just because it does not appear as you expected.
- This calculator performs integration calculations under the assumption that integrals are always positive, even when the integrals switch between positive and negative.

$\lceil f(x)$
$\lfloor F(x)$: primitive function of $f(x)$

$$\int_a^b f(x)dx = F(b) - F(a)$$