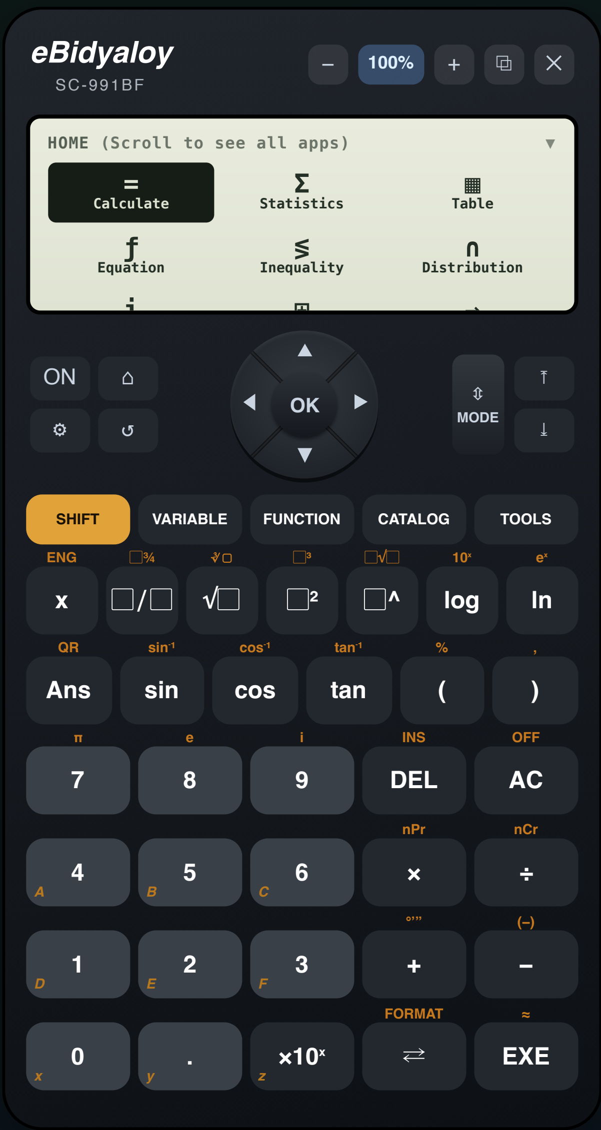
EBIDYALOY · SCIENTIFIC CALCULATOR EMULATOR

SC-991BF

Windows macOS Web Android iOS



Complete User's Guide & Function Reference — everything you need to operate the SC-991BF scientific calculator emulator, with every example result computed by the calculator engine.

Contents

About This Guide

Welcome

SC-991BF Emulator

The **eBidyaloy SC-991BF** is an on-screen scientific calculator emulator for classrooms and study — available on **Windows, macOS, Web, Android and iOS**. It reproduces a modern ClassWiz-style scientific calculator: a natural “textbook” display, 13 calculator apps, physical constants, unit conversions, and a live step-by-step explainer for teaching.

> Throughout this guide, every worked **Example** shows the expression, the calculator display, and the exact key operation. All results are produced by the actual SC-991BF calculation engine, so what you read here is what the emulator computes.

How to read the key operations

Key presses are shown as chips, e.g. 7 × 8 EXE. A key’s **SHIFT** function (orange label above the key) is written SHIFT then the key; a **VARIABLE** letter (gold) is entered from the VARIABLE menu.

Keys & Key Markings

The Faceplate

Zones

The SC-991BF keypad is arranged in the same zones as a modern scientific calculator: a title strip, the LCD, a control cluster, five soft keys, two scientific rows, and the number pad.

Key Markings — three functions per key

Most keys have up to three functions: **7** **Primary** — printed on the key; press it directly. **π** **SHIFT** — the orange label above the key; press **SHIFT** first, then the key. ***A*** **VARIABLE** — the gold letter; entered from the **VARIABLE** menu. For example, **SHIFT** **7** inputs **π**, and **SHIFT** **sin** inputs **sin⁻¹**.

Control Cluster

Key	Function
ON	Turns the calculator on and clears the current entry (AC also clears).

⌂ HOME	Opens the HOME screen — the menu of all calculator apps.
⚙ SETTINGS	Opens the SETTINGS menu (Calc Settings and Reset).
↺ Back	Deletes the character before the cursor, or steps back one menu level.
▲ ▼ ◀ ▶	Cursor keys — move the entry cursor, or move the highlight in menus and tables.
OK	Executes the calculation or selects the highlighted menu item (same as EXE).
⤒ ⤓	Page keys — jump to the top / bottom of a long result, menu, table or list.

Soft Keys

Soft key	Opens
SHIFT	Selects the alternate (orange) function printed above the next key you press.
VARIABLE	The variable menu — recall or store the memories A, B, C, D, E, F, x, y, z and M (M also has M+ / M–).
FUNCTION	A function menu for the current app (e.g. Abs, arg, Conjg, ReP, ImP for complex work).
CATALOG	The CATALOG of commands and functions, plus **CONST** ▸ (physical constants) and **CONV** ▸ (unit conversions).
TOOLS	App tools for the current screen. In **Calculate** it holds **Undo / Redo** for the last editing change; in the data apps (Statistics, Table, Distribution) it holds the row tools **Insert Row / Delete Row / Delete All** (Statistics adds **Sort Ascending / Descending**); in Spreadsheet it holds Fill / Copy / Grab and the display options. Result conversions (S⇔D, recurring decimal, ° ’ ”, improper fraction) live on the ⇄ (FORMAT) key and in the CATALOG.

Scientific Keys

Key	SHIFT	Function
x	ENG	Inserts the variable **x**. **SHIFT** converts the current result to engineering notation (ENG) — the same as FORMAT ▸ ENG Notation.
□/□	□¾	Fraction template a/b. SHIFT inserts a mixed-number template.
√□	∛□	Square root. SHIFT inserts a cube root.
□²	□³	Square (x^2). SHIFT cube (x^3).
□^	□√□	Power x^y. SHIFT the x-th root of a value.
log	10^x	Common logarithm (base 10). SHIFT raises 10 to a power.
ln	e^x	Natural logarithm (base e). SHIFT raises e to a power.
Ans	QR	Inserts the previous answer. **SHIFT** shares the current calculation as a QR code and link.
sin	$\sin^{-1}$	Sine. SHIFT arcsine (inverse).
cos	$\cos^{-1}$	Cosine. SHIFT arccosine (inverse).
tan	$\tan^{-1}$	Tangent. SHIFT arctangent (inverse).
(	%	Open parenthesis. SHIFT the percent operator.

)	,	Close parenthesis. SHIFT a comma (argument separator).

Number Pad

Key	SHIFT	VARIABLE	Function
7 8 9	π e i	—	Digits. SHIFT: 7→π, 8→e, 9→i (imaginary unit).
4 5 6	—	A B C	Digits, or recall variables A / B / C via VARIABLE.
1 2 3	—	D E F	Digits, or recall variables D / E / F via VARIABLE.
0 .	—	x y	Digit / decimal point, or variables x / y.
DEL	INS	—	Deletes the item left of the cursor. SHIFT (INS) arms the next function key (√, ∛, sin, cos, tan, ln, log) to **wrap the value to the right of the cursor** as its argument — e.g. cursor left of 30, INS, then √ gives √(30).
AC	OFF	—	All clear. SHIFT turns the calculator off.
×	nPr	—	Multiply. SHIFT permutation (nPr).
÷	nCr	—	Divide. SHIFT combination (nCr).
+	° ’ ”	—	Add. SHIFT sexagesimal (degrees-minutes-seconds) entry.
–	(–)	—	Subtract. SHIFT the negative sign for a signed value.
$\times 10^x$	—	z	Enters a power-of-ten exponent (scientific entry). Inserts a compact ᴇ mark or a **×10^** power expression depending on the **×10 Key** setting. VARIABLE: z.
⇄ (FORMAT)	—	—	S⇔D: toggles the result between decimal and exact (fraction / √ / π) form. Opens the FORMAT menu when set to do so.
EXE	≈	—	Executes the calculation. SHIFT forces a decimal (≈) result.

Menu-Operation Shorthand

To keep instructions short, this guide writes menu paths in a compact form. For example:

⚙ – [Calc Settings] > [Angle Unit] > [Degree]

…is the same as the full operation:

1. Press ⚙.
2. Use ▲▼ to select **[Calc Settings]**, then press OK.
3. Select **[Angle Unit]**, then press OK.
4. Select **[Degree]**, then press OK.

Note

Where a menu item shows an option number to its left, you can also press that number key to jump straight to the item — see **Using Menus**.

Entering & Editing Calculations

Entering a Calculation

Natural textbook input

The SC-991BF uses **natural textbook input** (MathI): fractions, roots, powers and other expressions appear on screen just as they are written on paper. You type an expression from left to right and press **EXE** to evaluate it. (By default results are shown in their exact "Standard" form — a fraction, surd or π; press **⇄** to switch to the decimal, or see Input/Output in SETTINGS to show decimals first.)

Keys such as **□/□** (fraction), **√□** (root) and **□^** (power) insert an empty **template** with boxes to fill. Type into the highlighted box, and press **▶** to move out of it and continue the calculation.

Example 1 To enter 1½ + √4

SHIFT **□/□** 1, 1, 2 **▶** **+** **√□** 4 **▶** **EXE**

```
Math DEG
1½ + √4
      3.5
```

The mixed-number template **SHIFT** **□/□** creates three boxes (whole, numerator, denominator); **▶** steps out of the root before **EXE**. The result 1.5 + 2 = 3.5 is shown.

Editing a Calculation

Cursor, insert & delete

Moving the cursor

Use the arrow keys to move the flashing cursor through an expression without erasing anything:

Key	Movement
◀ ▶	Move left / right one item along the current line.
▲ ▼	Move between levels of a template — for example up into the numerator of a fraction or down into the denominator.

Deleting and inserting

To correct a mistake, move the cursor to the spot and use:

Key	Action
DEL	Delete the item immediately to the **left** of the cursor.
SHIFT DEL (INS)	**Use the value to the right as an argument.** Arms the *next* function key so it wraps the value immediately right of the cursor — see below.
AC	Clear the whole expression and start again.

Example 2 To fix 7 × 9 typed as 7 × 8

◀ DEL 9 EXE

```
Math DEG
7 × 9
      63
```

With the cursor after the 8, press DEL to remove it, type 9, then EXE. Only the wrong digit is changed — the rest of the expression is untouched.

Using INS — wrap a value into a function

Sometimes you have already typed a value and then realise it should sit *inside* a function — for example you typed **30** but wanted **√30**. Instead of deleting and retyping, place the cursor to the **left** of the value, press SHIFT DEL (INS) to arm it, then press the function key. The function is inserted and the value to its right is pulled into the function's argument automatically.

INS arms these function keys: √□ (and ∛), sin, cos, tan (and their inverses), ln and log. The value it grabs is the next number, the next parenthesised group, or the next single template (fraction, root, variable…).

Example 3 To turn 30 into √30 without retyping

cursor left of 30 SHIFT DEL (INS) √□

```
Math DEG
√30
5.477225575
```

With the cursor before the 3, SHIFT DEL arms INS; pressing √□ inserts the radical and wraps 30 inside it, giving √30. Without arming, √□ would insert an empty √□ before the 30 instead.

> **Note**
> INS is a **one-shot** arm: it applies only to the very next function key you press, then clears. It has no effect on SHIFT that does not insert a wrapping function.

Correcting after a result

After you press EXE and see a result, you have two choices:

- Begin a **new** calculation — just start typing, and the previous entry is cleared.
- **Edit** the previous entry — press ◀ or ▶ to bring the expression back with the cursor in it, change what you need, and press EXE again.

When a calculation has an error

If you press EXE on an expression the calculator cannot evaluate, it shows an error message — most often **Syntax ERROR** or **Math ERROR**, sometimes a more specific name (see *Error Messages* in the appendix) — and returns you to the expression with the **cursor placed at the point that caused the error** — not at the end of the line. You can start fixing straight away without hunting for the fault. For example, an unmatched bracket or a stray operator leaves the cursor exactly where the parser stopped; a Math ERROR such as division by zero leaves it on the offending term.

Example 4 The cursor lands on the error

enter 5 + × 2 EXE → cursor sits on the ×

```
Math DEG
5 + × 2

Syntax ERROR
```

The doubled operator + × is invalid; EXE reports **Syntax ERROR** and drops the cursor on the ×. Press DEL to remove it and continue — you never have to retype the line.

Calculation History

Recall & replay

The calculator keeps your last **30** calculations. From a result, press ▲ to step back through previous entries and ▼ to step forward. A recalled entry can be re-used in two ways:

- Press EXE to **re-run** the recalled calculation as it is.
- Move the cursor into it, **edit** the expression, then press EXE to evaluate the changed version.

Example 5 To recall and re-run a previous calculation

at a result → ▲ browse → EXE re-run

```
Math DEG
7 × 8 − 4 × 5

36
```

Pressing ▲ brings back an earlier entry such as 7 × 8 – 4 × 5; EXE evaluates it again. Editing it first lets you try a variation without retyping the whole line.

Important!

Calculation history is cleared when you switch to a different app or reset the calculator. Use a **variable** (Store into A–F, x, y, z, M) to keep a value you will need after leaving the app.

Reusing the last answer with Ans

The result of the most recent calculation is held in **Ans**. Start a new calculation with an operator (for example + or ×) and the calculator inserts **Ans** automatically, or press Ans to use it anywhere in an expression.

Example 6 To chain calculations with Ans

1000 − 200 EXE → × 2 EXE

```
Math DEG
Ans × 2

1600
```

After 1000 – 200 = 800, pressing × 2 continues from that answer: Ans × 2 = 1600. This lets you build up a long calculation one step at a time.

Using Menus & Settings

Using Menus

Navigation

Many operations use menu screens. Press ⚙ while using a calculator app to open the SETTINGS menu shown below. Each item with a ▸ opens a further menu.

❶ Calc Settings ▸
❷ Reset ▸

Two ways to select an item:

- **Method 1** — use ▲ ▼ to move the highlight, then press OK.
- **Method 2** — press the number key shown to the left of the item (its **option number**). For example, press 2 to open **Reset** directly.

Press ↺ to step back one level, or AC to leave the menu.

Note

Option numbers (❶ ❷ ❸ …) let you jump to an item even before its menu is showing. Pressing ⚙ on the HOME screen opens Get Started instead of SETTINGS.

Using the SETTINGS Menu

Calc Settings & Reset

The SETTINGS menu has two groups: **Calc Settings** (how calculations behave and display) and **Reset**. Drill in with OK; each setting shows a numbered list of options to choose from.

Calc Settings
❶ Input/Output ▸
❷ Angle Unit ▸
❸ Number Format ▸
❹ Fraction Result ▸

Angle Unit
❶ Degree
❷ Radian
❸ Gradian

Calc Settings

Setting	Options	What it does
Input/Output	MathI/MathO · MathI/DecimalO	Natural textbook entry either way (fractions, roots and powers appear in two-dimensional form). **MathO** (the default) shows a result in its exact “Standard” form first — a fraction, a surd or π — and the ⇄ (S⇔D) key switches to the decimal value. **DecimalO** shows the decimal first instead, with ⇄ switching to the exact form. (Fix/Sci number formats always show a decimal.)
Angle Unit	Degree · Radian · Gradian	The angle unit used by trigonometric functions and their results.
Number Format	Norm 1 · Norm 2 · Fix 0–9 · Sci 1–10	How results are shown. **Norm** (the default is Norm 2) shows a decimal result to at most **6 decimal places**, trimming trailing zeros, and switches to exponential form outside a range (Norm 1 switches sooner). **Fix** fixes the number of decimal places (0–

		9 — choose Fix 7–9 to show more than six); **Sci** fixes the number of significant figures.
Fraction Result	Improp Fraction · Mixed Number	Whether a fractional answer shows as an improper fraction (7⁄3) or a mixed number (2 1⁄3).
Complex Result	a+bi · $r\angle\theta$	Whether a complex answer shows in rectangular (a + bi) or polar ($r\angle\theta$) form.
FORMAT Key	Decimal · Format Menu	What the ⇄ FORMAT key does — toggle straight to decimal, or open the FORMAT menu of display options.
×10 Key	Sci-Notation · Power	What the $\times 10^x$ key inserts — a compact **E** exponent mark (3E5) or a written-out **×10^** power expression (3×10^5). Both evaluate the same.
Digit Separator	On · Off	Thousands grouping in results, e.g. 1,234,567.
Decimal Mark	Dot · Comma	The decimal character and matching grouping: 1,234.5 (Dot) or 1.234,5 (Comma).

Example — set the angle unit to Degree: ⚙ – [Calc Settings] > [Angle Unit] > [Degree]. Trigonometric calculations now use degrees.

Reset

Reset item	What it does
Settings & Data	Returns all settings to their defaults and clears app data.
Variable Memory	Clears the variables A, B, C, D, E, F, x, y, z and M.
Initialize All	Resets everything — settings, memory and data.

Note

A reset cannot be undone. **Settings & Data** and **Initialize All** clear your work, so use them deliberately.

Using CATALOG, FUNCTION & TOOLS

Advanced-Command Soft Keys

CATALOG · FUNCTION · TOOLS

Three soft keys along the top of the keypad — CATALOG, FUNCTION and TOOLS — open on-screen menus that reach the calculator's advanced commands without dedicated keys. What each menu offers depends on the app you are in, so the same key does the right thing in Calculate, Complex, Spreadsheet and the rest. This chapter explains all three, with worked examples.

Note
Every result below is produced by the actual SC-991BF engine. Menu tags in the key path (grey pills such as [Func Analysis] or [SOLVE]) are the category you open and the item you highlight and insert with OK.

The CATALOG Menu

Commands, functions & symbols

Press CATALOG to open the commands, functions and symbols available in the current app. Following the Casio fx-991CW+ layout, the CATALOG is organised into **categories**: the first screen lists the categories, and opening one shows the items inside it. Move the highlight with ▲ ▼, press OK to open a category (or to insert / run the highlighted item), and press ↺ to step back a level or close.

```
CATALOG
  Func Analysis          ▸
  Probability            ▸
  Numeric Calc           ▸
  Angle/Coord/Sexa       ▸
  Hyperbolic/Trig        ▸
  Engineer Symbol        ▸
  Sci Constants ▸        ▸
  Unit Conversions ▸     ▸
  Other                  ▸
```

```
FUNC ANALYSIS
  d/dx derivative
  ∫□dx integral
  Σ summation
  Π product
  log□(□) log base
  log log₁₀
  ln natural log
```

Two levels. Choose a category on the left, press OK, and its items appear (right — **Func Analysis** shown). Most items are then **inserted** into your calculation with the caret ready for the first argument. A few — the special commands **SOLVE**, **CALC**, **FACT**, **ENG** and **VERIFY** — **act on what is already there** instead of being inserted. The **Sci Constants** ▸ and **Unit Conversions** ▸ categories open their own sub-menus (see below).

Note
Key paths in this chapter name both steps — for example CATALOG [Func Analysis] > [d/dx] means: open the CATALOG, open the **Func Analysis** category, then choose **d/dx**. Press the option number, or ▲ ▼ then OK, at each level.

CATALOG categories & items

Category	Items in the category
Func Analysis	d/dx derivative · ∫□dx integral · Σ summation · Π product · log□(□) log to any base · log ($\log_{10}$) · ln (natural log)
Probability	% percent · x! factorial · nPr (P) permutation · nCr (C) combination · Ran# random number · RanInt#(□,□) random integer
Numeric Calc	Abs |x| · Int truncate · Frac fractional part · Intg floor · Rnd round · GCD(□,□) · LCM(□,□)
Angle/Coord/Sexa	° ʳ ᵍ angle-unit marks · ° ′ ″ deg-min-sec input · Pol(□,□) rectangular→polar · Rec(□,□) polar→rectangular
Hyperbolic/Trig	sin · cos · tan · $\sin^{-1}$ · $\cos^{-1}$ · $\tan^{-1}$ · sinh · cosh · tanh · $\sinh^{-1}$ · $\cosh^{-1}$ · $\tanh^{-1}$
Engineer Symbol	m μ n p f (10^{3}…10^{15}) · k M G T P E (10^{3}…10^{18}) — insert an SI-prefix multiplier
Sci Constants ▸	Physical constants (opens the CONST sub-menu — six categories)
Unit Conversions ▸	Unit conversions (opens the CONV sub-menu — nine categories)
Other	Ans · π · e · √(· ⁿ√(· x^{-1} reciprocal · x^2 square · ^(power · (–) negative sign · comma · parentheses
Result Tools	FACT prime factorise · ENG / ENG→ engineering form · RECUR recurring decimal (output) · □̄ input recurring (type $0.\bar{3}$)
Matrix/Vector	MatA–MatD, MatAns · VctA–VctD, VctAns · det(□) · Trn(□) transpose · Identity(□)
Equation Tools	SOLVE f(x)=0 · CALC evaluate with variables · VERIFY test a relation
Symbols	: multi-statement · → store to variable · =, ≠, <, >, ≤, ≥ relations

Function Analysis

These build a calculus or algebra template with empty boxes; type into each box and move between them with ▶ or the D-pad, then press EXE.

Example 1 log to any base — $\log_2 32$

CATALOG Func Analysis log□(□) 2 ▶ 3 2 EXE

```
Math DEG
log₂ 32
      5
```

Enter the base in the small box, the number in the large box: $\log_2 32 = 5$, because $2^5 = 32$.

Example 2 Reciprocal — 4^{-1}

4 CATALOG Other x^{-1} EXE

```
Math DEG
4⁻¹
   0.25
```

x^{-1} (in the **Other** category) appends the reciprocal power to the value on the line: $4^{-1} = 1 \div 4 = 0.25$.

Example 3 Factorial — 5!

5 CATALOG Probability x! EXE

```
Math DEG
5!
 120
```

x! multiplies every whole number down to 1: 5! = 5·4·3·2·1 = 120.

Example 4 Derivative — d/dx(x^2) at x = 3

CATALOG Func Analysis d/dx x^2 ▶ 3 EXE

```
Math DEG
d/dx (x²)|₃
          6
```

The numerical derivative evaluates the slope of x^2 at x = 3, which is 2x = 6. In the Extended Display, the explainer shows the power-rule working (see the Explainer chapter).

Example 5 Integral — ∫ x^2 dx from 0 to 1

CATALOG Func Analysis ∫□dx x^2 ▶ 0 ▶ 1 EXE

```
Math DEG
∫₀¹ x² dx
0.3333333333
```

The definite integral is the area under x^2 between 0 and 1: 1⁄3 ≈ 0.3333.

Example 6 Summation — Σ x for x = 1 to 10

CATALOG Func Analysis Σ x ▶ 1 ▶ 1 0 EXE

```
Math DEG
Σ x  (1→10)
         55
```

Σ adds the expression as x runs over the range: 1 + 2 + … + 10 = 55.

Example 7 Product — Π x for x = 1 to 5

CATALOG Func Analysis Π x ▶ 1 ▶ 5 EXE

```
Math DEG
Π x  (1→5)
        120
```

Π multiplies instead of adds: 1·2·3·4·5 = 5! = 120.

Numeric Calc & Probability

Example 8 Greatest common divisor — GCD(1071, 462)

CATALOG Numeric Calc GCD(□,□) 1071 ▶ 462 EXE

```
Math DEG
GCD(1071, 462)
                21
```

The greatest common divisor of 1071 and 462 is 21.

Example 9 Least common multiple — LCM(4, 18)

CATALOG Numeric Calc LCM(□,□) 4 ▶ 18 EXE

```
Math DEG
LCM(4, 18)
          36
```

The least common multiple of 4 and 18 is 36.

Example 10 Random integer — RanInt#(1, 6) (a dice roll)

CATALOG Probability RanInt#(□,□) 1 ▶ 6 EXE

```
Math DEG
RanInt#(1, 6)
             4
```

Each press returns a fresh whole number between 1 and 6 — **your value will differ**. **Ran#** (no arguments) instead gives a random decimal in [0, 1).

Example 11 Rectangular → polar — Pol(3, 4)

CATALOG Angle/Coord/Sexa Pol(□,□) 3 ▶ 4 EXE

```
Math DEG
Pol(3, 4)
r = 5, θ = 53.13°
```

The point (3, 4) has magnitude r = 5 and angle θ = 53.13° (Degree mode). The pair is stored to x (= r) and y (= θ).

Example 12 Polar → rectangular — Rec(2, 60°)

CATALOG Angle/Coord/Sexa Rec(□,□) 2 ▶ 60 EXE

```
Math DEG
Rec(2, 60)
x = 1, y = 1.732
```

The polar point r = 2, θ = 60° becomes (1, 1.732); the pair is stored to x and y.

Example 13 Int, Frac & Intg — the parts of –3.5

CATALOG Numeric Calc Int –3.5) EXE

```
Math DEG
Int(−3.5)  Frac(−3.5)  Intg(−3.5)
                    −3 −0.5 −4
```

Int truncates toward zero (–3), **Frac** keeps the fractional part (–0.5), **Intg** takes the floor — the largest integer not above the value (–4).

Hyperbolic & trigonometric functions

Example 14 sinh 1

CATALOG Hyperbolic/Trig sinh 1) EXE

```
Math DEG
sinh 1
1.1752011936
```

sinh 1 = (e – e^{-1}) / 2 ≈ 1.1752. cosh, tanh, their inverses, and sin / cos / tan (+ inverses) are all in the **Hyperbolic/Trig** category.

Engineer Symbol — SI-prefix multipliers

```
ENGINEER SYMBOL
   m milli 10⁻³
   μ micro 10⁻⁶
   n nano 10⁻⁹
   k kilo 10³
   M mega 10⁶
   G giga 10⁹
```

The **Engineer Symbol** category **inserts** an SI prefix as a power-of-ten multiplier — exactly as Casio's fx-991CW+ does. Type a number, choose a prefix, and it acts like "×10^n": **5k** means 5 × 10^3 = 5000, **2m** means 2 × 10^{-3} = 0.002. The prefixes run **m *μ* n p f** (10^{-3} down to 10^{-15}) and **k M G T P E** (10^3 up to 10^{18}).

Note

This **inserts** a multiplier into the calculation — it is *not* the same as converting a finished result to engineering form. To reshape a *result* into ×10^n / SI-prefix form, use **ENG / ENG→** (Result Tools, the FORMAT menu, or SHIFT x).

Example 15 Engineer Symbol — 5 k = 5000

5 CATALOG Engineer Symbol k kilo 10³ EXE

```
Math DEG
5k
5000
```

Choosing **k** after 5 inserts the kilo multiplier: 5k = 5 × 10^3 = **5000**. It behaves like a constant, so it can be combined in a larger expression (e.g. 5k + 200).

Example 16 Engineer Symbol — 4.7 k Ω in a division

1 2 ÷ 4 . 7 CATALOG Engineer Symbol k kilo 10^3 EXE

```
Math DEG
12 ÷ 4.7k
0.002553…
```

4.7k = 4700, so 12 ÷ 4.7k = 12 ÷ 4700 ≈ **0.002553** — handy for electronics where values are written 4.7 kΩ, 10 μF, and so on.

Special commands (they act, not insert)

These CATALOG entries operate on the expression or result already on screen rather than inserting a template.

Example 17 SOLVE — a root of $x^2 - 4 = 0$

x $\square^2$ – 4 CATALOG Equation Tools SOLVE EXE

```
Math DEG
x² − 4 = 0
x = 2
```

SOLVE applies Newton's method from the stored value of x and finds the nearby root x = 2, storing it back into x.

Example 18 CALC — evaluate 2A + B with A = 3, B = 4

2 VARIABLE A + VARIABLE B CATALOG Equation Tools CALC

```
Math DEG
2A + B
10
```

CALC keeps the expression and prompts for each variable it contains — enter A = 3 then B = 4 — then evaluates 2·3 + 4 = 10. Press EXE on the result to re-enter values.

Example 19 FACT — prime-factorise 1080

1 0 8 0 EXE CATALOG Result Tools FACT

```
Math DEG
1080
2³ × 3³ × 5
```

FACT rewrites a whole-number result as its prime factors: 1080 = $2^3 \times 3^3 \times 5$.

Example 20 ENG — engineering notation for 12345

1 2 3 4 5 EXE CATALOG Result Tools ENG

```
Math DEG
12345
12.345 × 10³
```

ENG shifts the exponent to a multiple of three; ◀ ▶ step it further. **ENG→** shows the same value with an SI prefix instead (12.345 k). ENG is also on the FORMAT menu and on SHIFT x — see the Calculate chapter.

Example 21 RECUR — the repeating form of 1 ÷ 7

1 ÷ 7 EXE CATALOG Result Tools RECUR

```
Math DEG
1 ÷ 7
0.[142857]
```

RECUR shows the repeating block in brackets: 1 ÷ 7 = 0.[142857] = 0.142857142857…

Example 22 VERIFY — is 3 × 4 = 12 ?

3 × 4 CATALOG Equation Tools VERIFY = 1 2 EXE

```
Math DEG
3 × 4 = 12
True
```

VERIFY tests a relation built with =, ≠, <, >, ≤ or ≥ and reports **True** or **False**.

Defining your own functions — f(x) and g(x)

The Table app lets you register two functions f(x) and g(x), but you can also define them **right here in Calculate**. Type an expression in **x**, then choose **Define f(x)** (or **Define g(x)**) from the CATALOG. The line is stored, and **f(☐)** / **g(☐)** become callable from any app — no need to open the Table app. Definitions persist until you overwrite them or reset.

Example 23 Define f(x) = 2x + 1, then evaluate f(3)

2 x + 1 CATALOG Define f(x) → CATALOG f(☐) 3) EXE

```
Math DEG
f(3)
7
```

Typing 2x + 1 and choosing **Define f(x)** stores it (the display confirms “f(x) defined”). Then **f(☐)** with 3 inside gives f(3) = 2·3 + 1 = **7**.

Example 24 Compose defined functions — g(f(2)), with g(x) = x²

define g(x)=x² CATALOG g(☐) CATALOG f(☐) 2)) EXE

```
Math DEG
g(f(2))
25
```

With f(x) = 2x + 1 and g(x) = x², nesting the calls gives g(f(2)) = g(5) = 5² = **25**. A definition may call the other function; a self-referential loop is caught and stopped with a **Stack ERROR**.

Sci Constants — physical constants (CATALOG → Sci Constants ▸)

The **Sci Constants** ▸ category opens a menu of scientific constants (CODATA-2022) in six sub-categories. Choose a sub-category, then a constant, to insert its value into your calculation.

```
CONST
❶ Universal            ▸
❷ Electromagnetic      ▸
❸ Atomic & Nuclear     ▸
❹ Physico-Chem         ▸
```

For example, inserting **c** (speed of light) inserts 299 792 458. The complete list is in the **Constants & Conversions** reference chapter.

Unit Conversions — (CATALOG → Unit Conversions ▸)

The **Unit Conversions** ▸ category applies a unit conversion to the value on the display. First compute a value, then choose a category and a conversion; the result is replaced by the converted value.

```
UNIT CONVERT
❶ Length     ▸
❷ Area       ▸
❸ Volume     ▸
❹ Mass       ▸
```

Conversions cover Length, Area, Volume, Mass, Velocity, Pressure, Energy, Power and Temperature. The full list is in the reference chapter.

The FUNCTION Menu

Modulus, argument & complex parts

Press **FUNCTION** to insert the functions that describe a number's size, angle and parts. **Abs** works on any real or complex value; **arg**, **Conjg**, **ReP**, **ImP** and the polar operator ∠ are used with complex numbers. Highlight an item, press **OK** to insert it, and the caret lands inside its parentheses.

```
FUNCTION
  Abs |z|
  arg angle
  Conjg z̄
  ReP real part
  ImP imaginary part
  ∠ polar r∠θ
```

Abs |z| — magnitude · **arg** — angle · **Conjg** z̄ — sign-flip the imaginary part · **ReP** / **ImP** — real / imaginary part · ∠ — enter a number in polar form r∠θ.

Example 25 Abs — the absolute value |2 – 7|

FUNCTION **Abs** 2 **–** 7 **)** **EXE**

```
Math DEG
|2 − 7|
      5
```

Absolute value strips the sign: |–5| = 5. On a real number Abs is just its distance from zero.

Example 26 Abs — the modulus of a complex number |3 + 4i|

FUNCTION Abs 3 + 4 i) EXE

```
Math DEG
|3 + 4i|
        5
```

For a complex number Abs is the modulus $\sqrt{(3^2 + 4^2)} = 5$.

> **Note**
>
> **Abs** adapts to its argument: a real number → its distance from zero; a complex number → its modulus; a **vector** → its magnitude |v|; a **matrix** → a new matrix with the absolute value of every element (see the Matrix chapter).

Example 27 arg — the argument of 3 + 4i

FUNCTION arg 3 + 4 i) EXE

```
Math DEG
arg(3 + 4i)
  53.13°
```

arg returns the angle of the number in the complex plane, in the current Angle Unit: 53.13° in Degree mode.

Example 28 Conjg — the conjugate of 3 + 4i

FUNCTION Conjg 3 + 4 i) EXE

```
Math DEG
Conjg(3 + 4i)
     3 − 4i
```

Conjg flips the sign of the imaginary part: 3 + 4i becomes 3 – 4i.

Example 29 ReP & ImP — the parts of 3 + 4i

FUNCTION ReP 3 + 4 i) EXE

```
Math DEG
ReP(3+4i)   ImP(3+4i)
                  3 4
```

ReP returns the real part (3) and **ImP** the imaginary part (4).

> **Note**
>
> The FUNCTION items are also reachable in the Complex app, where the whole calculation is complex by default. The polar operator ∠ lets you type a number as r∠θ (for example 5∠53.13°).

The TOOLS Menu

App tools for the current screen

Press TOOLS to reach tools for the app you are in. TOOLS is **context-sensitive** — it holds editing and app operations, **not** result-format conversions. (Those live on the ⇄ FORMAT key and the CATALOG; see the table below.)

TOOLS in Calculate — Undo / Redo

```
TOOLS
  Undo reverse last edit
  Redo re-apply
```

On the Calculate screen the TOOLS menu holds **Undo** and **Redo**. Undo reverses your last editing change to the current expression — a mistaken DEL, a template you did not mean to insert, or a cleared line — and Redo re-applies it. This works on the expression you are editing, before you press EXE.

TOOLS in the data apps — row editing

In the list-based apps, TOOLS edits the data list. In **Statistics** you can also sort:

App	TOOLS menu offers
Calculate	**Undo** · **Redo** — reverse / re-apply the last editing change.
Statistics	**Insert Row** · **Delete Row** · **Delete All** · **Sort Ascending** · **Sort Descending** (sorts the highlighted column; a paired x/Freq or x/y list is kept aligned).
Table / Distribution	**Insert Row** · **Delete Row** · **Delete All** (no sort — row order is meaningful).
Spreadsheet	Edit Cell · Fill Formula · Fill Value · Copy & Paste · Cut & Paste · Grab (insert a cell reference) · Show Cell · Auto Calc (on/off) · Recalculate · Delete All.
Equation	Complex Roots — toggle whether roots with a negative discriminant are shown as complex numbers or reported as “No Real Roots”.

Where the result-format conversions live

Conversions such as S⇔D, recurring decimal, engineering form and prime factorisation are **not** on the TOOLS menu. Reach them here:

To…	Press
Toggle a result between exact ⇄ decimal (S⇔D)	⇄ (FORMAT) after a result
Open the FORMAT menu — Standard (π √ frac) · Decimal · Prime Factor · Recurring Decimal · Rectangular / Polar Coord · Improper / Mixed Fraction · Standard Form $a\times10^n$ · **ENG Notation** · Sexagesimal (only the items that suit the current result appear)	SHIFT ⇄, or ⇄ when the **FORMAT Key** setting is “Format Menu”
Engineering form — ENG / ENG→ (with SI symbol)	CATALOG [Result Tools] ▸ ENG / ENG→, the FORMAT menu ▸ **ENG Notation**, or the shortcut SHIFT x
Verify a relation is true	CATALOG [Equation Tools] ▸ VERIFY

Note

The Calculate chapter works through each of these conversions with numbered examples.

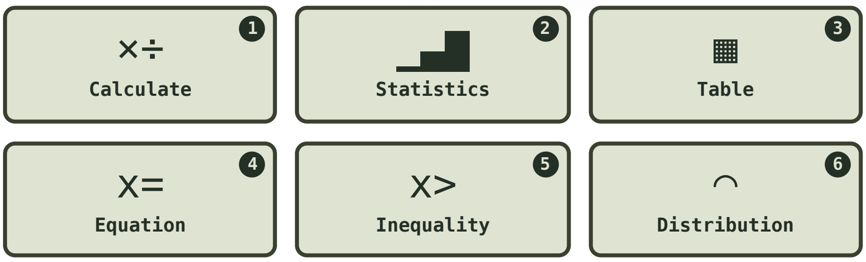
Calculator Apps

Selecting a Calculator App

HOME screen

Press ⌂ to display the HOME screen — the menu of all installed calculator apps. Use ▲ ▼ ◀ ▶ to move the highlight to an app, then press OK. Alternatively, press the number key shown on an app's icon (its option number) to open it directly.

The HOME screen on the SC-991BF — every installed app in a scrollable 3-across grid, the highlighted app (Calculate here) named along the bottom. Move the highlight with ▲▼◀▶ and press OK to open it.



i Complex 7	[▦] Matrix 8	→ Vector 9
⊞ Spreadsheet 10	a:b Ratio 11	⚄ Math Box 12
0x Base-N 13		

Note

Each app remembers its own screen, data and CATALOG. Switching apps does not clear another app's data.

Installed Calculator App List

13 apps

App	Description
×÷ Calculate 1	General and scientific calculations — arithmetic, functions, powers, roots, logs, complex numbers and every CATALOG command.
Statistics 2	1- and 2-variable statistics and seven regression models, with a data-entry table and a full set of summary results.
▦ Table 3	Generates a table of values from one or two functions, f(x) and g(x), over a start/end/step range.
X= Equation 4	Solves simultaneous linear equations (2 to 4 unknowns) and polynomial equations (quadratic, cubic, quartic).
X> Inequality 5	Solves quadratic, cubic and quartic inequalities and reports the solution intervals.
⌒ Distribution 6	Normal, Binomial and Poisson probability — both probability density (PD) and cumulative distribution (CD).
i Complex 7	Dedicated complex-number arithmetic with rectangular (a + bi) and polar (r∠θ) results.
[▦] Matrix 8	Matrix arithmetic up to 4×4 — addition, subtraction, multiplication, determinant, inverse and transpose.

9 → Vector	2D and 3D vector operations — dot product, cross product, magnitude, angle and unit vectors.
10 ⊞ Spreadsheet	A 5-column (A–E) × 45-row spreadsheet with cell formulas, fill/copy tools and range functions (Sum, Min, Max, Mean).
11 a:b Ratio	Solves proportions of the form a : b = c : x for the unknown term.
12 ⚄ Math Box	Probability and learning tools — dice roll, coin toss, number line and circle simulations.
13 0x Base-N	Binary, octal, decimal and hexadecimal calculations with logic operators (and, or, xor, not).

App Screens at a Glance

Entry screens

When you open an app you see either a calculation screen or a short menu to choose the calculation type. Each app is documented in full — every function with worked examples — in the chapters that follow.

×÷ 1 Calculate

APP 1

Calculate

General and scientific calculations — arithmetic, functions, powers, roots, logs, complex numbers and every CATALOG command.

```
Math DEG
                                        0
```

2 Statistics

APP 2

Statistics

1- and 2-variable statistics and seven regression models, with a data-entry table and a full set of summary results.

```
Statistics
❶ 1-Variable
❷ y=a+bx
❸ y=a+bx+cx²
```

▦ 3
Table

APP 3
Table

Generates a table of values from one or two functions, f(x) and g(x), over a start/end/step range.

```
Table
f(x)=
```

x= 4
Equation

APP 4
Equation

Solves simultaneous linear equations (2 to 4 unknowns) and polynomial equations (quadratic, cubic, quartic).

```
Equation
❶ Simult. Equation
❷ Polynomial
```

x> 5
Inequality

APP 5
Inequality

Solves quadratic, cubic and quartic inequalities and reports the solution intervals.

```
Inequality
❶ Order 2 (ax²…)
❷ Order 3
❸ Order 4
```

⌒ 6
Distribution

APP 6
Distribution

Normal, Binomial and Poisson probability — both probability density (PD) and cumulative distribution (CD).

```
Distribution
❶ Normal PD
❷ Normal CD
❸ Binomial PD
```

i 7
Complex

APP 7
Complex

Dedicated complex-number arithmetic with rectangular (a + bi) and polar (r∠θ) results.

```
Math Complex
                                        0
```

[▦] 8
Matrix

APP 8
Matrix

Matrix arithmetic up to 4×4 — addition, subtraction, multiplication, determinant, inverse and transpose.

```
Matrix
❶ Define MatA
❷ Define MatB
❸ MatA + MatB
```

→ 9
Vector

APP 9
Vector

2D and 3D vector operations — dot product, cross product, magnitude, angle and unit vectors.

```
Vector
❶ Define VctA
❷ Define VctB
❸ VctA · VctB
```

⊞ 10
Spreadsheet

APP 10
Spreadsheet

A 5-column (A–E) × 45-row spreadsheet with cell formulas, fill/copy tools and range functions (Sum, Min, Max, Mean).

```
Spreadsheet
A1
```

a:b 11
Ratio

APP 11
Ratio

Solves proportions of the form a : b = c : x for the unknown term.

```
Ratio
❶ a:b = c:x
❷ a:b = x:d
```

⚄ 12
Math Box

APP 12
Math Box

Probability and learning tools — dice roll, coin toss, number line and circle simulations.

```
Math Box
❶ Dice Roll
❷ Coin Toss
❸ Number Line
```

0x 13
Base–N

APP 13

Base-N

Binary, octal, decimal and hexadecimal calculations with logic operators (and, or, xor, not).

```
Base–N DEC
                                                   0
```

Extended Display & Explainer

The Extended Display

Presentation mode

When teaching with the SC-991BF on a large screen, open the **extended display** with the (cast) button in the title bar. A greatly enlarged copy of the calculator's LCD appears beside the keypad so the whole class can read the current expression and result. The panel is marked **LIVE DISPLAY** and refreshes automatically as you type.

> Use the size controls (– % +) to fit the calculator to any display, and press **HIDE** to collapse the panel. The extended display always mirrors exactly what is on the calculator's own screen.

The Explainer

EXPLANATION & STEPS

Below the extended display, the **explainer** turns any Calculate result into a lesson. It has two tabs:

- **EXPLANATION** — a plain-language walkthrough of what each function and operation does, in order.
- **STEPS** — the working shown as a reduction: the expression is rewritten one step at a time, each line a little more simplified, down to the final answer (shown in green).

> **Note**
> The explainer is available in the Calculate app for any expression you evaluate. Both the wording and the step chain are produced by the calculator itself — they always match the actual computation.

Example 1 — roots and arithmetic

eBidyaloy · EXTENDED DISPLAY • LIVE DISPLAY

```
Math DEG
2×√(2)+3
                5.8284271248
```

EXPLANATION | STEPS

EXPLANATION

A square root asks: which number, multiplied by itself, gives 2? Since 1.4142135624 × 1.4142135624 = 2, we get √2 = 1.4142135624.

Multiply 2 by 1.4142135624: 2 × 1.4142135624 = 2.8284271248.

Add 2.8284271248 and 3 together: 2.8284271248 + 3 = 5.8284271248.

STEPS

```
2×√(2)+3
√2 = 1.4142135624
2 × 1.4142135624 = 2.8284271248
2.8284271248 + 3 = 5.8284271248
```

Example 2 — trigonometry (Degree)

eBidyaloy · EXTENDED DISPLAY ● LIVE DISPLAY

Math DEG

sin(30)+cos(60)

1

EXPLANATION STEPS

EXPLANATION

sin gives sine of the angle. Here sin(30°) = 0.5.

cos gives cosine of the angle. Here cos(60°) = 0.5.

Add 0.5 and 0.5 together: 0.5 + 0.5 = 1.

STEPS

sin(30)+cos(60)

sin(30°) = 0.5

cos(60°) = 0.5

0.5 + 0.5 = 1

Example 3 — powers

eBidyaloy · EXTENDED DISPLAY ● LIVE DISPLAY

Math DEG

3^2+4^2

25

EXPLANATION STEPS

EXPLANATION

Squaring means multiplying a number by itself. So $3^2 = 3 \times 3 = 9$.

Squaring means multiplying a number by itself. So $4^2 = 4 \times 4 = 16$.

Add 9 and 16 together: 9 + 16 = 25.

STEPS

3^2+4^2

$3^2 = 9$

$4^2 = 16$

9 + 16 = 25

Example 4 — logarithms

eBidyaloy · EXTENDED DISPLAY ● LIVE DISPLAY

Math DEG

log(1000)+ln(1)

3

EXPLANATION STEPS

EXPLANATION

log gives logarithm (base 10). Here log(1,000) = 3.

In gives natural logarithm (base e). Here ln(1) = 0.

Add 3 and 0 together: 3 + 0 = 3.

STEPS

```
log(1000)+ln(1)
log(1,000) = 3
ln(1) = 0
3 + 0 = 3
```

Example 5 — implicit multiplication priority

eBidyaloy · EXTENDED DISPLAY ● LIVE DISPLAY

Math DEG

6÷2(1+2)

1

EXPLANATION STEPS

EXPLANATION

Divide 6 by 2 — splitting 6 into 2 equal parts: 6 ÷ 2 = 3.

Add 1 and 2 together: 1 + 2 = 3.

Multiply 3 by 3 (writing them side by side means multiply): 3 × 3 = 1.

STEPS

```
6÷2(1+2)
6 ÷ 2 = 3
1 + 2 = 3
3 × 3 = 1
```

Example 6 — order of operations

eBidyaloy · EXTENDED DISPLAY ● LIVE DISPLAY

Math DEG

7+3×4−2

17

EXPLANATION STEPS

EXPLANATION

Multiply 3 by 4: 3 × 4 = 12.

Add 7 and 12 together: 7 + 12 = 19.

Subtract 2 from 19: 19 − 2 = 17.

STEPS

```
7+3×4−2
3 × 4 = 12
7 + 12 = 19
19 − 2 = 17
```

Example 7 — a derivative, explained with the power rule

For a **d/dx(** derivative of a polynomial, the explainer doesn't just report the numeric slope — it reads the polynomial in x, applies the **power rule** $d/dx(x^n) = n \cdot x^{n-1}$ term by term to build the derivative function $f'(x)$, then substitutes the point to get the slope. (For a non-polynomial such as sin or ln it gives the rate-of-change / tangent-slope explanation instead.)

eBidyaloy · EXTENDED DISPLAY ● LIVE DISPLAY

Math DEG

d/dx(3x^2−5x+1) at x = 4

19

EXPLANATION STEPS

EXPLANATION

Differentiate f(x) = 3x^2−5x+1 with respect to x. Apply the power rule $d/dx(x^n) = n \cdot x^{n-1}$ to each term.

Bringing each exponent down and reducing it by one gives the derivative function.

Substitute x = 4 into $f'(x)$ to get the slope at that point.

STEPS

```
d/dx(3x^2−5x+1) at x = 4
f(x) = 3x^2−5x+1
f′(x) = 6x − 5
f′(4) = 19
```

Note

The result 19 is the engine's numeric derivative; the power-rule working shown here is the explainer confirming *why* $f'(4) = 6 \cdot 4 - 5 = 19$. Both come from the calculator itself, so they always agree.

Sharing Calculations (QR & Link)

Sharing a Calculation

QR code & link

Any result on the SC-991BF can be turned into a **QR code and a shareable web link**. A student scans the code (or opens the link) to see the calculation on a phone or computer — typeset neatly, with the step-by-step working and, where it applies, a graph. It is the fastest way to put a worked example on the board, into a worksheet, or into a chat with a student.

Press **SHIFT** **Ans** (the orange **QR** label above the **Ans** key) to share the calculation currently on screen. A panel shows the QR code and a copyable link.

Sharing works in every app

Sharing is not limited to the Calculate app. Press **SHIFT** **Ans** on the **result screen of any app** — Statistics, Equation, Matrix, Vector, Complex, Base-N, Distribution, Inequality, Ratio or Table — and that result is shared with its heading, its answer (matrices and tables included) and its working.

Note

Sharing is anonymous: no login is needed to open a shared link. The shared page is read-only — it shows the calculation but cannot change your calculator.

The Shared Page

What the link opens

Opening a shared link shows a clean, read-only page with the calculation rendered in proper mathematical notation. Depending on the app, the page can include:

- **Typeset math** — the expression and answer in textbook form (fractions, roots, powers, matrices).
- **A graph** — for **Table** (the function curve), **Statistics** (the scatter of points with the fitted regression line) and **Distribution** (the normal curve or probability bars).
- **Step by step** — the same teaching breakdown the calculator's explainer shows.
- **Institute name** — when shared from a signed-in institute account, the page shows which institute it came from.

Sample shared page (with a graph)

eB **eBidyaloy SC-991BF**
Shared calculation

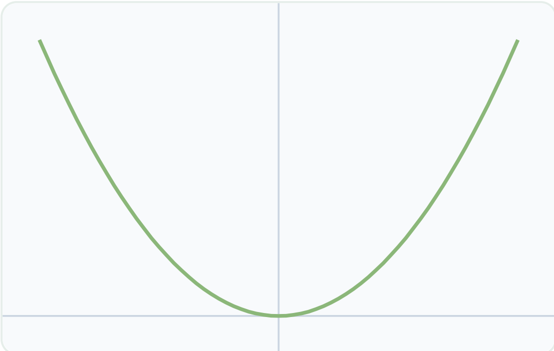
Table · Degree

Function Table f(x) = x^2

x	f(x)
−3	9
−2	4
−1	1

0	0
1	1
2	4
3	9

Graph



Step by step

Each row evaluates $f(x) = x^2$ at one x value. The points trace a symmetric U-shaped parabola with its vertex at (0, 0).

⌛ **Kept for 2 weeks — expires on 30 Jul 2026.**

Shared by **Dream Model Academy**

A shared **Table** result for $f(x) = x^2$. The page shows the value table, the plotted parabola, the step-by-step note, the institute that shared it, and the expiry date. This is exactly what a student sees after scanning the QR code — no app to install.

Important!

Shared calculations are kept on the server for **2 weeks**, then they are deleted automatically. The shared page always shows the exact date the link expires. Share a calculation again to get a fresh 2-week link.

Open in calculator

A shared Calculate link also offers **“Open in calculator”**, which reloads the exact expression back into the SC-991BF so it can be edited and re-run. This needs an eBidyaloy account with the calculator enabled; anyone else can still read the shared page.

Note

Available on Windows, macOS, Web and Android. The shared page opens in any modern browser on any device.

Calculate App Reference

Calculate

General & scientific calculation

The **Calculate** app is the calculator's main workspace. Enter an expression in natural textbook format and press EXE to evaluate it. This chapter covers every function available in Calculate, grouped by type, each with a worked example.

> **Note**
>
> Unless a note says otherwise, examples assume the default settings — Angle Unit: Degree, and MathI/MathO input/output. Under **MathO** (the default) a result appears in its exact “Standard” form first — a fraction, surd or π; press ⇄ (S⇔D) to switch to its decimal value. Where an example quotes a decimal, that is the ⇄ (decimal) form. Switch Input/Output to **DecimalO** to show the decimal first instead.

Basic Arithmetic & Priority

Use + − × ÷ for the four operations and ( ) to group terms. Calculations follow standard priority: functions and powers first, then × ÷, then + −. An omitted × (implicit multiplication) before a bracket or constant binds tighter than ÷.

Example 1 To calculate 7 × 8 – 4 × 5

7 × 8 − 4 × 5 EXE

```
Math DEG
7 × 8 − 4 × 5
                36
```

Multiplication is done before subtraction: 56 – 20 = 36.

Example 2 To calculate 6 ÷ 2(1 + 2)

6 ÷ 2 (1 + 2) EXE

```
Math DEG
6 ÷ 2(1 + 2)
                 1
```

Implicit multiplication binds tighter than ÷, so this reads 6 ÷ (2 × (1 + 2)) = 6 ÷ 6 = 1.

Fractions

Press □/□ for a fraction template and SHIFT □/□ for a mixed number. With the default MathO output a result appears as the exact fraction; press ⇄ (FORMAT / S⇔D) to switch to its decimal value (or select DecimalO to show the decimal first).

Example 3 To calculate 3⁄4 + 1⁄6

3 □/□ 4 ▶ + 1 □/□ 6 EXE

```
Math DEG
3⁄4 + 1⁄6
0.9166666667
```

The result is shown as the decimal 0.9166666667. Press ⇄ to see it in exact form as the fraction 11⁄12.

Powers & Roots

[□²] squares, [SHIFT] [□²] cubes, [□^] raises to any power, [√□] is a square root, [SHIFT] [√□] a cube root, and [SHIFT] [□^] gives the x-th root.

Example 4 To calculate $5^2 + 12^2$

5 [□²] [+] 1 2 [□²] [EXE]

```
Math DEG
5² + 12²
169
```

25 + 144 = 169. (Its square root, 13, is the hypotenuse of a 5-12-13 triangle.)

Example 5 To calculate the 5th root of 32

SHIFT [□^] 5 [▶] 3 2 [EXE]

```
Math DEG
⁵√32
2
```

The x-th root key gives $\sqrt[5]{32} = 2$, because $2^5 = 32$.

Example 6 To calculate 4^{-1} (reciprocal)

4 [CATALOG] [x⁻¹] [EXE]

```
Math DEG
4⁻¹
0.25
```

The reciprocal $\mathbf{x^{-1}}$ is 1 ÷ x, so $4^{-1} = 0.25$.

Exponential & Logarithmic

[log] is base-10 log, [ln] is natural log, [SHIFT] [log] is 10^x, [SHIFT] [ln] is e^x. Use the CATALOG **log□(□)** for a logarithm to any base.

Example 7 To calculate log 1000

log 1 0 0 0 [)] [EXE]

```
Math DEG
log 1000
3
```

log 1000 = 3, because $10^3 = 1000$.

Example 8 To calculate $\log_2 32$ (log to base 2)

CATALOG log□(□) → 2, 32 → EXE

```
Math DEG
log₂ 32
        5
```

Enter the base in the small box: $\log_2 32 = 5$, because $2^5 = 32$.

Example 9 To calculate ln e^3 (natural logarithm)

ln e □^ 3 ▶) EXE

```
Math DEG
ln e³
      3
```

ln is the natural logarithm (base e). It undoes e^x, so ln $e^3 = 3$.

Example 10 To calculate e^2 (e^x)

SHIFT ln 2) EXE

```
Math DEG
e²
7.3890560989
```

SHIFT ln inputs e^x (e raised to a power): $e^2 \approx 7.3891$.

Trigonometric Functions

sin cos tan and their inverses (SHIFT + the key) use the current Angle Unit. Append ° r g (from CATALOG) to give a value in a specific unit regardless of the mode.

Example 11 To calculate sin 30° (Degree mode)

sin 3 0) EXE

```
Math DEG
sin 30°
     0.5
```

In Degree mode sin 30° = 0.5.

Example 12 To calculate $\tan^{-1} 1$ (Degree mode)

SHIFT tan 1) EXE

```
Math DEG
tan⁻¹ 1
      45
```

The inverse tangent of 1 is the angle whose tangent is 1: 45°.

Hyperbolic Functions

The hyperbolic functions sinh, cosh, tanh and their inverses are on the CATALOG menu.

Example 13 To calculate sinh 1

CATALOG sinh 1) EXE

```
Math DEG
sinh 1
1.1752011936
```

sinh 1 = $(e - e^{-1}) / 2 \approx 1.1752$.

Percentage

Enter a value followed by SHIFT ((the % operator). A percentage is interpreted as “per hundred”.

Example 14 To calculate 150 × 20%

1 5 0 × 2 0 SHIFT (EXE

```
Math DEG
150 × 20%
30
```

20% of 150 is 30.

Permutation, Combination & Factorial

SHIFT × is nPr, SHIFT ÷ is nCr, and **x!** (factorial) is on the CATALOG menu.

Example 15 To calculate 5 P 2 (permutations)

5 SHIFT × 2 EXE

```
Math DEG
5 P 2
20
```

The number of ordered arrangements of 2 from 5 is 20.

Example 16 To calculate 8 C 3 (combinations)

8 SHIFT ÷ 3 EXE

```
Math DEG
8 C 3
56
```

The number of unordered selections of 3 from 8 is 56.

Example 17 To calculate 5! (factorial)

5 CATALOG x! EXE

```
Math DEG
5!
        120
```

The factorial **x!** multiplies every whole number down to 1: 5! = 5 × 4 × 3 × 2 × 1 = 120.

Numeric Functions

The CATALOG provides **Abs** (absolute value), **Int** (truncate), **Frac** (fractional part), **Intg** (floor), **Rnd** (round to the display) and $\mathbf{x^{-1}}$ (reciprocal).

Example 18 To calculate |2 – 7| (absolute value)

FUNCTION Abs 2 – 7) EXE

```
Math DEG
|2 – 7|
          5
```

The absolute value strips the sign: |–5| = 5.

Example 19 Int, Frac & Intg — the parts of –3.5

CATALOG Int –3.5) EXE

```
Math DEG
Int(–3.5)  Frac(–3.5)  Intg(–3.5)
                     –3 –0.5 –4
```

Int truncates toward zero (–3), **Frac** keeps the fractional part (–0.5), and **Intg** takes the floor — the largest integer not above the value (–4).

Example 20 Rnd — round off 2 ÷ 3

CATALOG Rnd 2 ÷ 3) EXE

```
Math DEG
Rnd(2 ÷ 3)
0.6666666667
```

Rnd rounds the value to the calculator's 10-digit display precision, discarding the internal guard digits it normally keeps.

GCD, LCM & Random

From the CATALOG: **GCD(**, **LCM(**, **RanInt#(** (a random integer in a range) and **Ran#** (a random number in [0, 1)).

Example 21 To calculate GCD(48, 36)

CATALOG GCD(48, 36 → EXE

```
Math DEG
GCD(48, 36)
12
```

The greatest common divisor of 48 and 36 is 12.

Example 22 To calculate LCM(6, 8)

CATALOG LCM(6, 8 → EXE

```
Math DEG
LCM(6, 8)
24
```

The least common multiple of 6 and 8 is 24.

Example 23 RanInt#(— a random integer from 1 to 6 (a dice roll)

CATALOG RanInt#(1, 6) EXE

```
Math DEG
RanInt#(1, 6)
4
```

Each press returns a fresh whole number between 1 and 6 — **your value will differ**. **Ran#** (no arguments) instead gives a random decimal in [0, 1).

Calculus — Derivative & Integral

The CATALOG offers **d/dx** (numerical derivative at a point) and **∫dx** (definite integral between limits).

Example 24 To calculate d/dx(x^2) at x = 3

CATALOG d/dx → x^2, at 3 → EXE

```
Math DEG
d/dx (x²)|₃
6
```

The slope of x^2 at $x = 3$ is $2x = 6$.

Example 25 To calculate ∫ x^2 dx from 0 to 1

CATALOG ∫dx → x^2, 0, 1 → EXE

```
Math DEG
∫₀¹ x² dx
0.3333333333
```

The area under x^2 from 0 to 1 is $1/3 \approx 0.3333$.

Summation & Product

Σ sums and **Π** multiplies an expression as a counter runs from a lower to an upper limit.

Example 26 To calculate Σ x for x = 1 to 10

CATALOG Σ → x, 1, 10 → EXE

```
Math DEG
Σ x  (1 → 10)
            55
```

1 + 2 + … + 10 = 55.

Example 27 To calculate Π x for x = 1 to 5

CATALOG Π → x, 1, 5 → EXE

```
Math DEG
Π x  (1 → 5)
          120
```

1 × 2 × 3 × 4 × 5 = 5! = 120.

SOLVE, CALC & VERIFY

SOLVE finds a root of f(x) = 0 by Newton's method. **CALC** evaluates an expression after prompting for each variable. **VERIFY** tests whether a relation (=, ≠, <, >, ≤, ≥) is true.

Example 28 SOLVE — a root of $x^2 - 4 = 0$

x □² – 4 CATALOG SOLVE EXE

```
Math DEG
x² − 4 = 0
     x = 2
```

SOLVE searches from the stored x and finds the nearby root x = 2.

> **Note**
>
> If SOLVE cannot converge within its iteration limit it shows the best value so far with the prompt **Could not converge · OK Continue · AC Exit**: press OK to iterate further from that value, or AC to keep the near-solution. (Same behaviour as the Equation app's Solver.)

Example 29 CALC — evaluate 2A + B with A = 3, B = 4

2 VARIABLE A + VARIABLE B CATALOG CALC

```
Math DEG
2A + B
    10
```

CALC prompts for A then B, then evaluates 2·3 + 4 = 10.

Example 30 VERIFY — is 3 × 4 = 12 ?

3 × 4 CATALOG VERIFY = 12 EXE

```
Math DEG
3 × 4 = 12
True
```

VERIFY confirms the relation is True.

Coordinate Conversion

Pol(converts rectangular (x, y) to polar (r, θ); **Rec(** converts polar (r, θ) back to rectangular (x, y). Both are on the CATALOG.

Example 31 Pol(— convert (3, 4) to polar

CATALOG Pol(3, 4 → EXE

```
Math DEG
Pol(3, 4)
r = 5, θ = 53.13°
```

The point (3, 4) has magnitude r = 5 and angle θ = 53.13° (Degree mode).

Example 32 Rec(— convert (2, 60°) to rectangular

CATALOG Rec(2, 60 → EXE

```
Math DEG
Rec(2, 60°)
x = 1, y = 1.732
```

The polar point r = 2, θ = 60° becomes (1, 1.732).

Variables & Independent Memory (M)

The calculator has ten memories — **A B C D E F x y z** and the independent memory **M** — that each hold one number. Press VARIABLE to open the menu, highlight a memory with ▲ ▼ and press OK; a small action menu then appears offering **Recall** or **Store** (M additionally offers **M+** and **M–**).

```
VARIABLE
A
B
C
D
E
F
x
y
z
M
```

```
A
Recall
Store
```

- **Store** evaluates whatever is on the current line — or reuses **Ans** if the line is empty — and saves the value into the chosen memory, replacing what was there.
- **Recall** inserts the memory's stored value into the calculation you are editing. Recalling straight after a result begins a fresh line.
- **M+ / M–** add or subtract the current value to/from **M** without overwriting it — ideal for accumulating a running total.

Clear all ten memories at once with ⚙ – [Reset] > [Variable Memory].

To store a value into a memory (for example, save 5 into A):

1. Type the value — **5** — on the line. *(Leave the line empty to store the last answer, **Ans**, instead.)*
2. Press VARIABLE to open the memory list.
3. Highlight the memory (**A**) with ▲ ▼ and press OK.
4. Choose **Store**. The value is now held in A until you overwrite it or reset the memory.

To recall a stored value: press VARIABLE, highlight the memory, press OK, then choose **Recall** — the stored value drops straight into whatever you are calculating.

Example 33 Store 5 into the variable A

5 VARIABLE A Store

```
Math DEG
5
        5
```

The value 5 is now held in A; the display confirms 5. It stays stored until you overwrite it or reset the memory.

Example 34 Recall A and calculate A × 3

VARIABLE A Recall × 3 EXE

```
Math DEG
A × 3
      15
```

Recalling A (= 5) gives 5 × 3 = 15.

Example 35 Running total with M+ : 3 × 4 then 5 × 6

3 × 4 VARIABLE M M+ 5 × 6 VARIABLE M M+

```
Math DEG
5 × 6
M = 42
```

Each M+ adds the line's value to M — first 12, then +30. Recall M (VARIABLE ▸ M ▸ Recall) to read the total 42.

> **Note**
>
> M+ / M– act on the current line's value. To add a value into M without displaying it first, type the value then choose VARIABLE ▸ M ▸ M+.

Storing a result from another app

You are not limited to storing from Calculate. On the **result** screen of the **Equation** solver, **Distribution**, **Vector**, **Complex** or **Matrix** app, press VARIABLE and the displayed value is captured; pick a memory (A–F, x, y, z) to store it. This lets you carry a solved root, a probability, a dot product or a determinant straight into a follow-up calculation without copying it down.

Example 36 Store an equation root into A, then reuse it

Equation result x = 3 VARIABLE A Store

```
Math DEG
x² − 5x + 6 = 0  →  A
                     3
```

Solving $x^2 - 5x + 6 = 0$ gives x = 3; on the result screen VARIABLE ▸ **A** stores 3 into A. Back in Calculate, A is now 3 and can be used anywhere. (A matrix result stores into MatA–MatD instead — see the Matrix chapter.)

> **Note**
>
> The value stored is the one on screen. On a list result (Distribution, or simultaneous equations) it stores the **highlighted** row / first variable; scroll to the value you want before pressing VARIABLE.

Ans, π, e & Constants

Ans reuses the previous answer; SHIFT 7 / 8 insert π and e. Physical constants come from CATALOG → CONST ▸.

Example 37 To calculate π × 2

SHIFT 7 × 2 EXE

```
Math DEG
π × 2
6.2831853072
```

π × 2 ≈ 6.2832. Press ⇄ to keep the exact form 2π.

Standard Form (a × 10^n)

Press ×10^x to type a power-of-ten exponent directly — the natural way to enter very large or very small numbers. To have *results* shown in standard form, set ⚙ – [Calc Settings] > [Number Format] > [Sci].

Example 38 To enter 1.5 × 10^4

1 . 5 ×10^x 4 EXE

```
Math DEG
1.5 × 10⁴
15,000
```

The ×10^x key enters the "× 10 to the power" part in a single keystroke: 1.5 × 10^4 = 15000.

The **×10 Key** setting (⚙ ▸ Calc Settings) chooses what that key inserts. With **Sci-Notation** it inserts a compact ᴇ exponent mark; with **Power** it inserts a written-out **×10^** expression. Both evaluate identically.

Example 39 Sci-Notation mode — the compact ᴇ mark

×10 Key ▸ Sci-Notation 3 ×10^x 5 EXE

```
Math DEG
3ᴇ5
300000
```

Here ×10^x inserts ᴇ: 3ᴇ5 means 3 × 10^5 = **300000**. This matches how scientific results are displayed.

Example 40 Power mode — a written-out ×10^ expression

×10 Key ▸ Power 3 ×10^x 5 EXE

```
Math DEG
3×10^5
300000
```

In Power mode the same key inserts **×10^**, so the entry reads 3×10^5 on screen — clearer for teaching standard form. The value is the same **300000**.

Result Formats & Tools

After a result, ⇄ (FORMAT / S⇔D) toggles exact ⇄ decimal, and the full **FORMAT** menu (SHIFT ⇄) offers Prime Factor, Recurring Decimal, Sexagesimal, Standard Form, **ENG Notation** and the fraction forms — showing only the ones that suit the current result. Engineering form also has the shortcut SHIFT x, and the CATALOG (Result Tools) adds ENG→, FACT and RECUR. (The TOOLS key here is Undo / Redo — see the CATALOG chapter.)

Example 41 S⇔D — toggle √8 between exact and decimal

√8) EXE ⇄

Math DEG
√8
$2\sqrt{2} \rightleftarrows 2.8284271248$

The exact form $2\sqrt{2}$ and the decimal 2.8284… are the same value shown two ways.

Example 42 FACT — prime-factorise 360

3 6 0 EXE CATALOG FACT

Math DEG
360
$2^3 \times 3^2 \times 5$

FACT breaks a whole number into its prime factors.

Example 43 ENG — engineering notation for 12345

1 2 3 4 5 EXE SHIFT x

Math DEG
12345
12.345×10^3

ENG shifts the exponent to a multiple of 3, ready for SI prefixes. The quickest route is the shortcut SHIFT x; it is also on the FORMAT menu (**ENG Notation**) and in the CATALOG (Result Tools ▸ ENG).

Example 44 ENG→ — the same value with an SI prefix

1 2 3 4 5 EXE CATALOG ENG→

Math DEG
12345
12.345 k

ENG→ replaces the $\times 10^n$ part with its SI prefix, so 10^3 becomes **k**: 12345 = **12.345 k**. Prefixes run from a (10^{-18}) through m, k, M … up to E (10^{18}); ◀ ▶ step the exponent.

Example 45 Recurring Decimal — the repeating form of 1 ÷ 7

1 ÷ 7 EXE SHIFT ⇄ Recurring Decimal

Math DEG
1÷7
0.[142857]

FORMAT ▸ **Recurring Decimal** shows the repeating block in brackets: 1 ÷ 7 = **0.[142857]** = 0.142857142857… (The CATALOG command **RECUR** does the same.)

Typing a recurring decimal

You can also **enter** a recurring decimal directly, using the CATALOG command $\bar{\square}$ **input recurring**. It inserts an overline block; type the repeating digits inside it, then press ▶ to leave the block. The calculator uses the exact value in the calculation.

Example 46 To enter $0.\overline{3}$ (= 1⁄3)

0 . CATALOG ▯ input recurring 3 EXE

```
Math DEG
0.3̄
0.3333333333
```

With no digits before the block, the whole fractional part repeats: $0.\overline{3}$ = **1⁄3** = 0.3333333333.

Example 47 A block after ordinary digits — $0.1\overline{6}$ (= 1⁄6)

0 . 1 CATALOG ▯ input recurring 6 EXE

```
Math DEG
0.16̄
0.1666666667
```

Digits typed *before* the block are not repeated — only the 6 recurs: $0.1\overline{6}$ = **1⁄6** = 0.1666666667. A block can hold several digits, so $0.\overline{142857}$ = 1⁄7.

Example 48 Sexagesimal — enter and read degrees-minutes-seconds

1 °′″ 3 0 °′″ 0 °′″ EXE

```
Math DEG
1°30′0″
1.5
```

The °′″ key (SHIFT + +) enters a sexagesimal value: 1°30′0″ equals **1.5**. On a decimal result, FORMAT ▸ **Sexagesimal** converts the other way (1.5 → 1°30′0″).

The FORMAT menu — context-sensitive result conversions

All result conversions live on the **FORMAT** menu, opened with SHIFT ⇄ (or a bare ⇄ when the **FORMAT Key** setting is "Format Menu"). Each item reshapes the *current result* without retyping it. The menu is **context-sensitive** — like the Casio fx-991CW+, it lists only the conversions that make sense for the result on screen, so you never scroll past greyed-out options. A whole number offers **Prime Factor**; a repeating decimal offers **Recurring Decimal**; a fraction offers **Improper / Mixed**; a complex result offers **Rectangular / Polar Coord**; and **Standard**, **Decimal**, **ENG Notation**, **Standard Form** and **Sexagesimal** are always available.

```
FORMAT
  Standard (π √ frac)
  Decimal
  Prime Factor
  Recurring Decimal
  Improper Fraction
  Mixed Number
  Standard Form a×10ⁿ
  ENG Notation
  Sexagesimal
```

Standard ⇄ **Decimal** is the ⇄ toggle; **Prime Factor**, **Recurring Decimal** and **Sexagesimal** match the CATALOG commands above. **ENG Notation** puts the result in engineering form (also on SHIFT x). **Rectangular** ⇄ **Polar Coord** appear only for a complex result (see the Complex chapter). The fraction items convert between fraction shapes; Standard Form gives a × 10^n. (The screen above shows the items active for a plain decimal — the coordinate and fraction items are hidden here.)

Example 49 ENG Notation on the FORMAT menu — 12345

1 2 3 4 5 **EXE** SHIFT **⇄** **ENG Notation**

```
Math DEG
12345
12.345 × 10³
```

FORMAT ▸ **ENG Notation** shifts the exponent to a multiple of three: 12345 = **12.345 × 10^3**. The same conversion is on the shortcut **SHIFT** **x**.

Example 50 Improper ⇄ Mixed number — 13⁄4

1 3 **□/□** 4 **EXE** SHIFT **⇄** **Mixed Number**

```
Math DEG
13⁄4  ⇄  3 1⁄4
3 1⁄4
```

A result shown as the improper fraction 13⁄4 becomes the mixed number **3 1⁄4** under FORMAT ▸ **Mixed Number**; FORMAT ▸ **Improper Fraction** turns it back. (The **⇄** key still toggles fraction ⇄ decimal.)

Example 51 Standard Form — convert a result to a × 10^n

1 2 3 4 5 6 **EXE** SHIFT **⇄** **Standard Form**

```
Math DEG
123456
1.23456 × 10⁵
```

FORMAT ▸ **Standard Form** rewrites the displayed value 123456 as **1.23456 × 10^5** — the scientific-notation form — without changing the number. This converts an existing result, whereas the **×10^x** key is for *entering* such a value.

Statistics App Reference

Statistics

1-variable & 2-variable analysis

The **Statistics** app summarises a set of data and fits regression models. You type values into a list, then read off statistics such as the mean, standard deviation and quartiles; for paired data it also finds the line or curve of best fit. Open it from HOME by pointing to **Statistics** and pressing **OK**.

Choosing an analysis type

```
Select Calculation
1 1-Variable
2 y = a + bx
3 y = a + bx + cx²
4 y = a + b·ln(x)
5 y = a·e^(bx)
```

Choose **1-Variable** for a single list of numbers, or one of the **2-Variable** regression models for paired (x, y) data. The list continues with $\mathbf{y = a \cdot b^x}$, $\mathbf{y = a \cdot x^b}$ and **y = a + b/x**. Changing the type later clears the data.

Entering data

Type a value and press **EXE** to drop to the next row. Move with the arrow keys, overwrite a cell by typing over it, and delete a value with **DEL**. Press **TOOLS** on the data editor for the row tools — **Insert Row**, **Delete Row**, **Delete All** and **Sort Ascending / Descending** (see "Editing and extending the data list" below). To weight values by how often they occur, turn on the **Frequency** column from ⚙ → **Statistics** ▸ **Frequency** ▸ **On**.

1-Variable statistics

Example 1 **To summarise the data 2, 5, 6, 8, 9**

Statistics ▸ 1-Variable → enter data → **FUNCTION**

1-VARIABLE RESULT (1/2)	
$\bar{x}$	6
Σx	30
Σx²	210
Σx³	1,590
Σx⁴	12,594
σ_x	2.4494897428
s_x	2.7386127875

The mean is $\bar{x}$ = Σx / n = 30 / 5 = **6**. The result also lists the running sums of powers Σx, Σx^2, Σx^3, Σx^4 used by the formulas. Two standard deviations are reported: $\boldsymbol{\sigma_x}$ (population, ÷ n) and $\mathbf{s_x}$ (sample, ÷ n – 1); their squares are the variances $\boldsymbol{\sigma^2_x = 6}$ and $\mathbf{s^2_x = 7.5}$, also shown on the list.

Enter the data with:

2 › **EXE** › **5** › **EXE** › **6** › **EXE** › **8** › **EXE** › **9** › **EXE** › **FUNCTION**

Example 2 To read the order statistics (same data)

scroll the result with ▼

```
1-VARIABLE RESULT (2/2)
minX        2
Q1        3.5
Median      6
Q3        8.5
maxX        9
```

Scrolling down shows the five-number summary: minimum 2, first quartile 3.5, median 6, third quartile 8.5 and maximum 9.

Note

σ (divide by n) treats the data as the whole population; s (divide by n – 1) treats it as a sample estimating a larger population. Both are always shown.

Example 3 To weight values with a frequency column

⚙ Frequency ▸ On → enter x and Freq

1-Variable

x	Freq
10	2
20	5
30	3

Here the value 10 occurs twice, 20 five times and 30 three times (n = 10 in total), giving $\bar{x}$ = **21**, σ_x = **7** and s_x = **7.3786…**.

Normal distribution from 1-variable data

On the 1-variable result screen, press **EXE** to open the **Norm-Dist** tools. They convert a data value x to a probability using the fitted mean and σ: **P(t)** is the lower-tail area Φ(t), **Q(t)** = Φ(t) – 0.5, **R(t)** = 1 – Φ(t), and **▸t** standardises x into t = (x – $\bar{x}$) / σ_x.

Example 4 To standardise x = 8 and read its probability

1-Variable Result → EXE → ▸t / P(

```
NORM-DIST
▸t (x=8)   0.8164965809
P(t)        0.792891967
Q(t)        0.292891967
R(t)        0.207108033
```

The value 8 standardises to t = (8 – 6) / 2.449 = 0.8165, and about **79.3%** of a normal population lies below it (P(t)).

Two-variable regression

Choose a 2-variable model to get an **x** and a **y** column. Enter each pair, then press **FUNCTION** for the fitted coefficients and the correlation r.

Example 5 To fit a line to (1,3) (2,5) (3,7) (4,8) (5,11)

Statistics ▸ $y = a + bx$ → enter pairs → FUNCTION

```
REGRESSION RESULT
a                 1.1
b                 1.9
r        0.9904434668
x̄                   3
ȳ                 6.8
```

The line of best fit is $\mathbf{y = 1.1 + 1.9x}$ and the correlation $\mathbf{r = 0.990}$ shows a strong positive relationship.

Example 6 To estimate $\hat{y}$ and $\hat{x}$ from the fitted line

VARIABLE ▸ $\hat{y}$ / $\hat{x}$

```
ESTIMATED VALUES
ŷ (x = 6)            12.5
x̂ (y = 9)    4.1578947368
```

Type a value then the estimate function: $\hat{y}$ predicts y on the line at x = 6 (12.5); $\hat{x}$ solves the line for x when y = 9.

Example 7 To fit an exponential model y = a·e^(bx)

Statistics ▸ y = a·e^(bx) → (1,2)(2,4)(3,8)(4,16) → FUNCTION

```
EXP REGRESSION
a                   1
b        0.6931471806
r                   1
```

The data doubles each step, so the fit is y = 1·e^(0.6931x) — and e^0.6931 = 2 exactly. Non-linear models drop any point that transforms to an undefined value.

Example 8 To fit a quadratic $y = a + bx + cx^2$ to (1,1) (2,4) (3,9) (4,16) (5,25)

Statistics ▸ $y = a + bx + cx^2$ → enter pairs → FUNCTION

```
QUADRATIC REGRESSION
a                   0
b                   0
c                   1
```

The points lie exactly on $y = x^2$, so the fit returns a = 0, b = 0, c = **1**. The quadratic model reports the three coefficients a, b, c (it does not report a correlation r).

Example 9 To fit a power model $y = a \cdot x^b$ to (1,2) (2,8) (3,18) (4,32)

Statistics ▸ $y = a \cdot x^b$ → enter pairs → FUNCTION

```
POWER REGRESSION
a                    2
b                    2
r                    1
```

These points are exactly $y = 2x^2$, so the power fit recovers a = **2**, b = **2** with a perfect correlation r = 1. The power, ab-exponential and inverse models are fitted the same way — pick the shape that matches your data.

Menu item	Model	Reports r?
$y = a + bx$	Linear	Yes
$y = a + bx + cx^2$	Quadratic	No (a, b, c)
$y = a + b \cdot \ln(x)$	Logarithmic	Yes
$y = a \cdot e^{bx}$	e Exponential	Yes
$y = a \cdot b^x$	ab Exponential	Yes
$y = a \cdot x^b$	Power	Yes
$y = a + b/x$	Inverse	Yes

The other regression models

The logarithmic, ab-exponential and inverse models are fitted exactly like the ones above — choose the shape that matches your data, enter the pairs, and press FUNCTION. Each reports its own a, b (and correlation r).

Example 10 Logarithmic fit $y = a + b \cdot \ln(x)$ to (1,2) (2,4) (3,5) (4,6)

Statistics ▸ $y = a + b \cdot \ln(x)$ → enter pairs → FUNCTION

```
LOGARITHMIC REGRESSION
a         1.9953857485
b         2.8377294683
r         0.9989026513
```

The best-fit curve is $\mathbf{y = 1.9954 + 2.8377 \cdot \ln(x)}$ with a very strong correlation r = 0.999. Use this model when y rises quickly then flattens.

Example 11 ab-exponential fit $y = a\cdot b^x$ to (1,6) (2,18) (3,54) (4,162)

Statistics ▸ $y = a\cdot b^x$ → enter pairs → FUNCTION

```
AB-EXP REGRESSION
a                    2
b                    3
r                    1
```

The data triples each step, so the fit is exactly $y = 2\cdot 3^x$ (a = 2, b = 3) with r = 1. The ab-exponential form multiplies by a constant factor b each step; the e-exponential form (Example 7) uses base e instead.

Example 12 Inverse fit y = a + b/x to (1,7) (2,4) (4,2.5)

Statistics ▸ y = a + b/x → enter pairs → FUNCTION

```
INVERSE REGRESSION
a                    1
b                    6
r                    1
```

These points sit exactly on **y = 1 + 6/x** (a = 1, b = 6, r = 1). The inverse model suits data that falls off like 1/x and levels toward a constant a.

Switching the model without re-entering data

All seven 2-variable models use the same (x, y) columns, so you can try a different fit on the **same data** without retyping it. On the regression result screen press FUNCTION to open **Select Reg Type**, choose a model (▲▼ and OK, or the number keys 1–7), and the calculator re-fits the entered pairs and shows the new coefficients. This makes it easy to compare, say, a linear against a quadratic fit and read off which has the better correlation.

Example 13 To re-fit the same data as a quadratic — Select Reg Type

2-var result FUNCTION Select Reg Type ▸ $y = a + bx + cx^2$ OK

```
RE-FIT · QUADRATIC
a                  1.6
b         1.4714285714
c         0.0714285714
r                    —
```

Starting from the linear fit of Example 5, FUNCTION ▸ **Select Reg Type** ▸ quadratic re-fits the very same five pairs as $y = a + bx + cx^2$, now reporting a third coefficient **c**. Switching between **1-Variable** and **2-Variable**, however, still clears the list (the columns differ).

The full 2-variable statistics list

Beyond the fit coefficients, a 2-variable result lists every summary statistic — the running sums, both standard deviations for x and y, and the data ranges. Scroll the result with ▼ to reach them.

Example 14 **The complete read-out for the line data (1,3) (2,5) (3,7) (4,8) (5,11)**

Regression Result → scroll ▼

2-VARIABLE RESULT (SUMS)	
n	5
Σx	15
Σy	34
Σx^2	55
Σy^2	268
Σxy	121
Σx^3	225
Σx^2y	489
Σx^4	979

These are the running sums the regression uses. Scrolling further lists the means **$\bar{x}$ = 3**, **$\bar{y}$ = 6.8**; the population deviations **σx = 1.4142135624**, **σy = 2.7129319932**; the sample deviations **sx = 1.5811388301**, **sy = 3.0331501776** (with variances σ^2x, σ^2y, s^2x, s^2y as their squares); and the ranges **minX = 1, maxX = 5, minY = 3, maxY = 11**. (Same data as Example 5.)

Editing and extending the data list

You can revise the data at any time without leaving the app. Move the cursor with the arrow keys and:

- **Correct a value** — point to the cell, type the new value and press EXE; it replaces the old one.
- **Clear a cell** — empty it with DEL. A blank row is simply skipped when the statistics are calculated, so you don't have to shuffle the rows below.
- **Add more data** — scroll to the first empty row at the bottom of the list and keep typing.

To reshape the list itself, press TOOLS on the data editor for the **row tools**:

- **Insert Row** — push a blank row in *above* the highlighted row (so you can add a value you missed between two others).
- **Delete Row** — remove the highlighted row and close the gap.
- **Delete All** — clear the whole list to start over.
- **Sort Ascending / Descending** — reorder on whichever column the cursor is in (x, y or Freq). A paired list is kept aligned, so each x keeps its own frequency or y partner. Note that a sort cannot be undone — the original order is not recoverable.

Press FUNCTION again after any change and every statistic is recomputed from the current list.

Note

You can also copy a single cell's value into a memory: on the Statistics Editor, put the cursor on the cell and press VARIABLE, then pick a variable (A–F, x, y, z) and choose Store. Handy for pulling one data point into another calculation.

Example 15 To correct a mistyped value (the 6 was entered as 60)

point to the wrong cell → 6 **EXE** → **FUNCTION**

1-Variable
x
2
5
6
8
9

Overwriting cell x_3 with 6 and pressing **FUNCTION** refreshes the whole result — no need to re-enter the other four values.

Example 16 To insert a missing value and then sort the list

point to a row → **TOOLS** Insert Row type 4 → **TOOLS** Sort Ascending

1-Variable
x
2
4
5
6
8
9

Highlighting the row that held 5 and choosing **Insert Row** opens a blank line above it; typing 4 fills it. **Sort Ascending** then reorders the whole column to 2, 4, 5, 6, 8, 9. With a Freq or y column present, each partner value moves with its x.

Statistics calculation screen

The **Statistics Calc** screen (**TOOLS** ▸ **Stat-Calc** from a result) is a free-form editor in which you can recall any statistical value by name and combine it with ordinary arithmetic. Press **VARIABLE** to pick the value to insert. The available values are:

Group	Values you can recall
Summation	n · Σx · Σx² · Σx³ · Σx⁴ (paired data adds Σy · Σy² · Σxy · Σx²y)
Mean / Var / Dev	x̄ (paired adds ȳ) · σx · σ²x · sx · s²x (paired adds σy · σ²y · sy · s²y)
Min / Max / Quartile	min(x) · Q_1 · Med · Q_3 · max(x) (paired: min(x) · max(x) · min(y) · max(y))
Regression (paired)	a · b · r (and c for a quadratic fit)

Example 17 To combine statistics in an expression ($\bar{x} \times 2 + \Sigma x$)

TOOLS ▸ Stat-Calc → **VARIABLE** inserts $\bar{x}$, Σx …

```
Stat DEG
x̄ × 2 + Σx
            42
```

Press **TOOLS** on a result screen for this editor. Using the 1-variable data ($\bar{x} = 6$, $\Sigma x = 30$): $\bar{x} \times 2 + \Sigma x = 12 + 30 =$ **42**. Any statistic can be reused this way.

Example 18 To recall a regression coefficient in the Stat-Calc screen

2-variable data TOOLS ▸ Stat-Calc → **VARIABLE** a / b / r

RECALLED COEFFICIENTS	
a	1.1
b	1.9
r	0.9904434668

With the line data from Example 5, **VARIABLE** recalls the fitted coefficients individually — here a = 1.1, b = 1.9, r = 0.990 — so you can feed them straight into a further calculation (for example b × 10 or a prediction a + b·7).

Estimating x from a quadratic fit

For a linear model, estimating x from a y value gives a single $\hat{x}$ (Example 6). A **quadratic** model is not one-to-one, so a given y can correspond to **two** x values — the calculator returns both, $\hat{x}_1$ and $\hat{x}_2$.

Example 19 To estimate x when y = 9 on the quadratic fit $y = x^2$

VARIABLE ▸ $\hat{x}$ → 9 → **EXE**

ESTIMATED X	
$\hat{x}_1$ (y = 9)	3
$\hat{x}_2$ (y = 9)	−3

Using the quadratic fit from Example 8 ($y = x^2$), the value y = 9 is reached at both **x = 3** and **x = −3**, so the estimate returns the pair $\hat{x}_1 = 3$, $\hat{x}_2 = -3$. When the two roots coincide, or the quadratic term c is zero, a single value is returned instead.

Statistical calculation formulas

For reference, these are the formulas the app uses. Sums run over all n data values (weighted by the Frequency column when it is on).

Mean $\bar{x} = \dfrac{\Sigma x}{n}$ $\bar{y} = \dfrac{\Sigma y}{n}$

Population SD $\sigma_x = \sqrt{\left(\dfrac{\Sigma(x - \bar{x})^2}{n}\right)}$ **Sample SD** $s_x = \sqrt{\left(\dfrac{\Sigma(x - \bar{x})^2}{n - 1}\right)}$

Variance σ^2_x and s^2_x are the squares of the deviations above.

Linear regression $y = a + bx$:

$$b = \frac{n \cdot \Sigma xy - \Sigma x \cdot \Sigma y}{n \cdot \Sigma x^2 - (\Sigma x)^2} \qquad a = \frac{\Sigma y - b \cdot \Sigma x}{n} = \bar{y} - b \cdot \bar{x}$$

$$r = \frac{n \cdot \Sigma xy - \Sigma x \cdot \Sigma y}{\sqrt{[(n \cdot \Sigma x^2 - (\Sigma x)^2)(n \cdot \Sigma y^2 - (\Sigma y)^2)]}}$$

Estimates $\hat{y} = a + bx \quad \hat{x} = \frac{y - a}{b}$

Quadratic regression $y = a + bx + cx^2$ — the coefficients a, b, c come from the least-squares normal equations, and x is estimated from a y value by solving the quadratic:

$$\hat{x} = \frac{-b \pm \sqrt{(b^2 - 4c(a - y))}}{2c} \quad \text{(two roots } \hat{x}_1, \hat{x}_2\text{)}$$

Note

The non-linear models (logarithmic, exponential, ab-exponential, power, inverse) are fitted by applying the same linear formulas to the transformed data — for example the log model fits $a + b \cdot X$ with $X = \ln x$, then maps back.

Distribution App Reference

Distribution

Normal, Binomial & Poisson

The **Distribution** app evaluates the common probability distributions. Choose a distribution from the menu, enter its parameters, and read the probability — or, for the list distributions, a whole table of probabilities.

Choosing a distribution

```
Distribution
1 Normal PD
2 Normal CD
3 Inverse Normal
4 Binomial PD
5 Binomial CD
```

PD gives the density / mass at a point; **CD** gives the cumulative probability up to a bound. The list also has **Poisson PD** and **Poisson CD**.

Normal distribution

Example 1 Normal PD — density f(x) at x = 0.5, μ = 0, σ = 1

Distribution ▸ Normal PD → x, μ, σ → EXE

```
NORMAL PD
x                 0.5
μ                   0
σ                   1
f(x)     0.3520653268
```

f(x) is the height of the bell curve, not a probability. For the standard normal the height at x = 0.5 is **0.3521**.

Example 2 Normal CD — P(–1 ≤ X ≤ 1) for μ = 0, σ = 1

Distribution ▸ Normal CD → Lower, Upper, μ, σ → EXE

```
NORMAL CD
Lower              -1
Upper               1
P        0.6826894723
```

The area between the bounds is the probability. This is the classic “68% within one σ”.

Example 3 Normal CD — heights ~N(170, 6), P(160 ≤ X ≤ 180)

Normal CD → 160, 180, 170, 6 → EXE

NORMAL CD	
Lower	160
Upper	180
μ	170
σ	6
P	0.9044193359

About **90.4%** of this population lies between 160 and 180 cm. For a one-sided probability use a very large or very small bound (e.g. Lower = -1×10^{99}).

Example 4 Inverse Normal — the value with 97.5% below it

Distribution ▸ Inverse Normal → Area, μ, σ → EXE

INVERSE NORMAL	
Area	0.975
μ	0
σ	1
xInv	1.9599639861

Inverse Normal reverses the question. The value with 97.5% of the standard-normal area below it is **1.96** — the familiar 95%-confidence cut-off.

Binomial distribution (list input)

Enter **N** (trials) and **p** (success probability) once, then a growable list of x values; a probability is returned for each. **PD** gives P(X = x); **CD** gives P(X ≤ x).

Example 5 Binomial PD — N = 10, p = 0.5, for x = 0…4

Distribution ▸ Binomial PD → N=10, p=0.5, x-list → EXE

Binomial PD

x	P
0	0.0009765625
1	0.009765625
2	0.0439453125
3	0.1171875
4	0.205078125

Ten fair coin tosses: the probability of exactly 3 heads is **0.1172**. Add as many x values as you like to build the whole table.

Example 6 Binomial CD — P(X ≤ 3), N = 10, p = 0.5

Binomial CD → N=10, p=0.5, x=3 → EXE

Binomial CD

x	P
3	0.171875

The cumulative form adds P(X = 0…3) = **0.171875** — the chance of at most 3 heads.

Poisson distribution (list input)

Example 7 Poisson PD — λ = 3, for x = 0…4

Distribution ▸ Poisson PD → λ=3, x-list → EXE

Poisson PD

x	P
0	0.0497870684
1	0.1493612051
2	0.2240418077
3	0.2240418077
4	0.1680313557

Events at an average rate λ = 3. The most likely counts are 2 and 3, each with probability **0.2240**.

Example 8 Poisson CD — P(X ≤ 2), λ = 3

Poisson CD → λ=3, x=2 → EXE

Poisson CD

x	P
2	0.4231900811

Adding P(X = 0, 1, 2) gives **0.4232** — the chance of two or fewer events.

Valid parameter ranges

Each distribution accepts parameters only within its natural range; a value outside the range produces a **Math ERROR**. Re-enter a value inside the range to continue.

Distribution	Parameter constraints
Binomial PD / CD	$0 \le p \le 1$, $N \ge 0$ (integer), $0 \le x \le N$
Poisson PD / CD	$\lambda \ge 0$, $x \ge 0$ (integer)
Normal PD / CD	$\sigma > 0$
Inverse Normal	$0 < \text{Area} < 1$, $\sigma > 0$

Example 9 An out-of-range parameter is rejected (Inverse Normal, Area = 1.5)

Inverse Normal → Area = 1.5 → **EXE**

```
INVERSE NORMAL
Area                 1.5
μ                      0
σ                      1
              Math ERROR
```

An area must be a probability between 0 and 1, so 1.5 gives **Math ERROR**. The same happens for a Normal $\sigma \le 0$ or a Binomial p outside [0, 1] — correct the value and press **EXE** again.

Important!

List distributions accept several x values at once — add rows to the x list and every row is evaluated, giving a probability table you can scroll.

Note

Press **TOOLS** on the x list for the row tools — **Insert Row** (add an x value between two others), **Delete Row** and **Delete All**. The list keeps its order, so there is no Sort here.

Table App Reference

Table

Function value tables

The **Table** app tabulates one or two functions over a range of x — ideal for plotting points, spotting where a function crosses zero, and comparing two functions side by side.

Defining the function(s) and range

Type the formula for **f(x)** using the **x** key for the variable. To tabulate a second function, turn on **g(x)** from ⚙ → **Table ▸ f(x)/g(x)**. The calculator then asks for **Start**, **End** and **Step** and builds a row for each x (up to **45 rows** for a single function, or **30 rows** when both f(x) and g(x) are tabulated).

Example 1 To tabulate $f(x) = x^2 - 3$ for x = 1 … 5

Table → $f(x) = x^2 - 3$ → Start 1, End 5, Step 1 → EXE

Table

x	f(x)
1	–2
2	1
3	6
4	13
5	22

Between x = 1 and x = 2 the value changes sign (–2 → 1), so the graph crosses zero there — a quick way to locate a root. Scroll with ▲▼.

Enter the function with:

x › x² › – › 3 › EXE › 1 › EXE › 5 › EXE › 1 › EXE

Example 2 To tabulate $f(x) = x^2 - 3$ and $g(x) = 2x + 1$ together

⚙ f(x)/g(x) ▸ On → $g(x) = 2x + 1$ → x = 1 … 4

Table

x	f(x)	g(x)
1	–2	3
2	1	5
3	6	7
4	13	9

The two columns are closest near x = 3 (6 vs 7), showing roughly where the curves $x^2 - 3$ and $2x + 1$ intersect.

> **Note**
>
> The row limit is **45** for a single function but only **30** when g(x) is also on (as in Example 2). If Start, End and Step would produce more rows than the limit, the list is truncated to fit. Edit the function or range from the ⚙ menu and the table regenerates.

Note

On the generated table, press TOOLS for the row tools — **Insert Row**, **Delete Row** and **Delete All** — to prune or add individual x rows without changing the Start/End/Step range.

Registering & using f(x) and g(x)

Defining a function in the Table app also **registers** it. Once **f(x)** (and optionally **g(x)**) is entered, you can call it by name in the **Calculate** app: insert **f(□)** or **g(□)** from the CATALOG and type a value for x in the box. This lets you reuse a formula without retyping it.

Note

You don't have to open the Table app to register a function — in **Calculate**, type an expression in x and choose CATALOG ▸ **Define f(x)** (or **Define g(x)**). See “Defining your own functions” in the CATALOG chapter.

Example 3 Evaluate a registered function — f(3), where f(x) = 2x + 1

Calculate CATALOG f(□) 3) EXE

```
Math DEG
f(3)
      7
```

With f(x) = 2x + 1 registered in the Table app, f(3) = 2·3 + 1 = 7.

Example 4 Composite function — g(f(2)), where f(x) = 2x + 1 and g(x) = x^2

Calculate CATALOG g(□) → f(□) 2 →)) EXE

```
Math DEG
g(f(2))
     25
```

The inner call gives f(2) = 5, then g(5) = 5^2 = 25. You can also make a function *itself* composite — define **g(x) = f(x)^2** — and the calculator resolves the nested call (a built-in guard stops any accidental circular reference with a **Stack ERROR**).

Note

Data retention: f(x) and g(x) stay registered when you switch between apps, so define them once in Table and use them in Calculate. Registering new functions replaces them; ⚙ – [Reset] > [Variable Memory] clears them.

Equation App Reference

Equation

Simultaneous systems & polynomials

The **Equation** app solves two kinds of problem: **simultaneous linear systems** with 2, 3 or 4 unknowns, and **polynomial equations** of degree 2, 3 or 4. You enter only the coefficients and the calculator returns the complete solution set.

Choosing an equation type

Equation Type
1 Simultaneous · 2 unknowns
2 Simultaneous · 3 unknowns
3 Simultaneous · 4 unknowns
4 Polynomial · ax^2+bx+c
5 Polynomial · ax^3+…

The list continues with **Polynomial · $ax^4+bx^3+cx^2+dx+e$** and a free-form **Solver · f(x)=0**. Point to a type and press **OK**; the calculator lays out exactly the coefficient boxes that type needs (or, for the Solver, a single expression editor).

Simultaneous equations

Each row of the grid is one equation. For 2 unknowns the columns are **x**, **y** and the constant **=**. Enter the coefficients left to right, pressing **EXE** after each.

Example 1 To solve 2x + 3y = 8 and x – y = –1

Equation ▸ Simultaneous · 2 unknowns → coefficients → EXE

SOLUTION	
x	1
y	2

Enter the two rows (2, 3, 8) and (1, –1, –1). The unique solution is **x = 1, y = 2**.

Key operation:

2 › EXE › 3 › EXE › 8 › EXE › 1 › EXE › (–) › 1 › EXE › (–) › 1 › EXE

Example 2 To solve a 3-unknown system

Equation ▸ Simultaneous · 3 unknowns → EXE

SOLUTION	
x	1
y	2
z	3

For x + y + z = 6, x – y + 2z = 5 and 2x + y – z = 1 the solution is **x = 1, y = 2, z = 3**. A 4-unknown system adds a w column.

> **Note**
>
> If a system has no unique solution the calculator reports **Infinite Solutions** or **No Solution** instead of values.

Polynomial equations

Enter the coefficients from the highest power down. The quadratic **$x^2 - 5x + 6 = 0$** uses a = 1, b = –5, c = 6.

Example 3 To solve the quadratic $x^2 - 5x + 6 = 0$

Equation ▸ **Polynomial · ax^2+bx+c** → **1, –5, 6** → **EXE**

ROOTS	
x_1	3
x_2	2

The two roots are **x = 3** and **x = 2**. Scrolling down also shows the vertex of the curve — see “Minimum / maximum of the curve” below.

Example 4 To solve the cubic $x^3 - 6x^2 + 11x - 6 = 0$

Polynomial · ax^3+… → **1, –6, 11, –6** → **EXE**

CUBIC ROOTS	
x_1	3
x_2	1
x_3	2

Three real roots: **1, 2 and 3**.

Example 5 To solve the quartic $x^4 - 5x^2 + 4 = 0$

Polynomial · ax^4+… → **1, 0, –5, 0, 4** → **EXE**

QUARTIC ROOTS	
x_1	2
x_2	1
x_3	–1
x_4	–2

Four real roots: **±1 and ±2**. Enter 0 for any missing power (here b and d are 0).

Minimum / maximum of the curve

After you solve a quadratic or cubic, the result screen also reports the **turning point(s)** of the curve y = f(x): a quadratic has a single vertex (its minimum if a > 0, maximum if a < 0); a cubic has a local maximum **and** a local minimum when its shape has them.

Example 6 The vertex of the quadratic $x^2 - 5x + 6$

Polynomial · ax^2+bx+c → 1, –5, 6 → solve → scroll ▼

```
ROOTS + VERTEX
x1            3
x2            2
Min x       2.5
Min y     -0.25
```

Below the two roots the screen shows the vertex. For $x^2 - 5x + 6$ ($a = 1 > 0$) it is a **minimum** at (x, y) = **(2.5, –0.25)** — the lowest point of the parabola, halfway between the roots.

Example 7 The local max & min of the cubic $x^3 - 3x$

Polynomial · ax^3+… → 1, 0, –3, 0 → solve → scroll ▼

```
LOCAL EXTREMA
Max x        -1
Max y         2
Min x         1
Min y        -2
```

The cubic $x^3 - 3x$ turns twice: a local **maximum** at (–1, 2) and a local **minimum** at (1, –2). A cubic with no turning points (a monotonic curve) instead reports **No Local Max/Min**.

Complex roots

When a polynomial has non-real roots, the **Complex Roots** toggle decides whether they are shown. It is **Off** by default (only real roots appear). Turn it on from the Equation **TOOLS** menu → **Complex Roots ▸ On**.

Example 8 To show the complex roots of $x^2 + 2x + 5 = 0$

TOOLS ▸ Complex Roots ▸ On → solve

```
COMPLEX ROOTS
x1       -1 + 2i
x2       -1 - 2i
```

This equation has no real roots. With the setting on it returns the conjugate pair **–1 ± 2i**; with it off it reports **No Real Roots**.

Solver — one equation in x

The **Solver** is the last item on the type menu. Instead of coefficients you type a whole expression **f(x)** (or an equation **L = R**), and the calculator finds a value of x that makes it zero using Newton's method, starting from the current value of x.

Example 9 To solve $x^3 - 2x - 5 = 0$ with the Solver

Equation ▸ Solver · f(x)=0 → $x^3 - 2x - 5$ → EXE

```
SOLVER
x            2.0945514815
L − R                   0
```

The Solver converges to **x = 2.0945514815**. The **L – R** line is the residual f(x) at that root — here 0 to display precision — showing how exactly the equation is satisfied. Type f(x) with the x key; to solve an equation such as 2x = x + 5, insert "=" from the CATALOG and enter both sides.

> **Note**
> The Solver returns the one root nearest the starting guess. To reach a different root, store a new starting value into x (VARIABLE ▸ x ▸ Store) before solving; to get every root of a polynomial at once, use the Polynomial types instead.

Example 10 When the Solver cannot converge — Continue or Exit

solve a hard f(x) → OK Continue / AC Exit

```
SOLVER
x              1.732050807
L − R          −3.2 × 10⁻⁶
Could not      OK Continue · AC
converge       Exit
```

If Newton's method has not settled within its iteration limit, the Solver does not just give up — it shows the **best value so far**, its residual **L – R**, and the prompt **Could not converge · OK Continue · AC Exit**. Press OK to run more iterations starting from that best value (repeat until L – R is acceptably small), or AC to **Exit** and keep the near-solution. A residual close to zero means the value is already a good approximate root.

> **Note**
> On the Solver result screen you can also **store the root**: press VARIABLE and pick a memory (A–F, x, y, z). See "Storing a result from another app" in the Calculate chapter.

Inequality App Reference

Inequality

Polynomial inequalities

The **Inequality** app solves a polynomial inequality and reports the solution as a set of intervals on the number line. It handles quadratic, cubic and quartic inequalities against 0.

Choosing degree and direction

```
Inequality Degree
1 Quadratic ax²+bx+c
2 Cubic ax³+bx²+cx+d
3 Quartic ax⁴+…+e
```

After the degree, choose the direction against zero — **> 0**, **< 0**, **≥ 0** or **≤ 0** — then enter the coefficients exactly as in the Equation app.

Example 1 To solve $x^2 - 5x + 6 > 0$

Inequality ▸ Quadratic ▸ > 0 → 1, –5, 6 → EXE

```
Ineq DEG
x² − 5x + 6 > 0
x < 2 or x > 3
```

The factors are $(x - 2)(x - 3)$, so the expression is positive **outside** the roots: $x < 2$ or $x > 3$.

Example 2 To solve $x^2 - 5x + 6 \le 0$

Quadratic ▸ ≤ 0 → 1, –5, 6 → EXE

```
Ineq DEG
x² − 5x + 6 ≤ 0
2 ≤ x ≤ 3
```

With ≤, the boundary roots are included, so the answer is the closed interval **$2 \le x \le 3$**. The app writes ≤ / ≥ for inclusive directions and < / > for strict ones.

Example 3 To solve $x^3 - x > 0$

Cubic ▸ > 0 → 1, 0, –1, 0 → EXE

```
Ineq DEG
x³ − x > 0
−1 < x < 0 or x > 1
```

The roots are –1, 0 and 1; testing each region gives two solution intervals, joined with **"or"**.

> **Note**
> If the inequality is true everywhere the app shows **All Real Numbers**; if it is never true it shows **No Solution**.

Complex App Reference

Complex

Arithmetic with complex numbers

The **Complex** app performs the four operations on two complex numbers $z_1 = a + bi$ and $z_2 = c + di$, and can show the result in rectangular (a + bi) or polar (r∠θ) form. The imaginary unit i satisfies $i^2 = -1$.

Selecting the operation

Operation
1 $z_1 + z_2$
2 $z_1 - z_2$
3 $z_1 \times z_2$
4 $z_1 \div z_2$

Pick the operation first, then enter the real and imaginary parts of each number into the coefficient grid. The examples below all use $z_1 = 3 + 2i$ and $z_2 = 1 + 4i$.

Example 1 **To add (3 + 2i) + (1 + 4i)**

Complex ▸ $z_1 + z_2$ → 3, 2, 1, 4 → EXE

RESULT
$z_1 + z_2$ 4 + 6i

Add the real parts and the imaginary parts separately: (3 + 1) + (2 + 4)i = 4 + 6i.

Example 2 **To subtract (3 + 2i) – (1 + 4i)**

$z_1 - z_2$ → EXE

RESULT
$z_1 - z_2$ 2 − 2i

Subtract componentwise: (3 – 1) + (2 – 4)i = 2 – 2i.

Example 3 **To multiply (3 + 2i) × (1 + 4i)**

$z_1 \times z_2$ → EXE

RESULT
$z_1 \times z_2$ −5 + 14i

Use the distributive law with $i^2 = -1$: (3·1 – 2·4) + (3·4 + 2·1)i = **–5 + 14i**. Enter it with 3 › EXE › 2 › EXE › 1 › EXE › 4 › EXE › FUNCTION

Example 4 **To divide (3 + 2i) ÷ (1 + 4i)**

$z_1 \div z_2$ → EXE

RESULT
$z_1 \div z_2$ 0.6470588235 − 0.5882352941i

Multiply top and bottom by the conjugate of the denominator (1 – 4i). The denominator becomes $1^2 + 4^2 = 17$, a real number.

Polar form and complex functions

Set **Complex Result ▸ r∠θ** in SETTINGS to read a result in polar form. The **FUNCTION** menu inside the app also offers **Abs** (modulus |z|), **arg** (argument), **Conjg** (conjugate $\bar{z}$), **ReP** (real part) and **ImP** (imaginary part).

Example 5 To read 1 + i in polar form

SETTINGS ▸ Complex Result ▸ r∠θ

```
POLAR (DEG)
1 + i                 √2 ∠ 45°
|1 + i|           1.4142135624
arg(1 + i)                  45
```

1 + i has modulus r = √2 ≈ 1.4142 and argument θ = 45° (in Degree mode). Changing the Angle Unit changes how θ is reported.

Example 6 To enter a number in polar form — 2∠60°

FUNCTION ▸ ∠ → 2 ∠ 60 → EXE

```
POLAR INPUT (DEG)
2∠60°          1 + 1.7320508076i
```

The ∠ operator on the **FUNCTION** menu lets you type a complex number by magnitude and angle as r∠θ. In Degree mode 2∠60° converts to rectangular **1 + 1.7320508076i** (= 1 + √3·i). You can mix polar and rectangular values in the same expression.

Note

Conjg flips the sign of the imaginary part (a + bi → a – bi); ReP and ImP extract the real and imaginary components as ordinary numbers.

Powers and functions in the Calculate app

The Complex app is a guided $z_1 \circ z_2$ workspace. For anything richer — **powers** of a complex number, longer expressions, or the **FUNCTION** functions applied on their own — use the **Calculate** app, where you type i with the **i** key and enter a free expression. Complex results there obey the same **FUNCTION** menu (Abs, arg, Conjg, ReP, ImP) and the SETTINGS r∠θ output.

Example 7 Conjg, ReP and ImP of 2 + 3i (Calculate app)

Calculate FUNCTION ▸ Conjg / ReP / ImP → 2 + 3i → EXE

```
COMPLEX FUNCTIONS
Conjg(2 + 3i)           2 − 3i
ReP(2 + 3i)                  2
ImP(2 + 3i)                  3
```

Conjg gives the conjugate 2 – 3i; **ReP** returns the real part 2 and **ImP** the imaginary part 3 as ordinary numbers.

Example 8 A power of a complex number — $(1 + i)^4 + (1 - i)^2$

Calculate → (1 + i) **□^** 4 + (1 − i) **□^** 2 → **EXE**

```
COMPLEX POWER
(1+i)⁴ + (1−i)²                -4 − 2i
```

Raise a complex number to a whole-number power with the **□^** key: $(1 + i)^4 = -4$ and $(1 - i)^2 = -2i$, so the sum is **−4 − 2i**. The exponent may be any integer.

Base-N App Reference

Base-N

Binary, octal, decimal & hexadecimal

The **Base-N** app works with whole numbers in four bases — **DEC** (10), **HEX** (16), **OCT** (8) and **BIN** (2) — and includes the bitwise logic operators. All arithmetic is integer arithmetic on a 32-bit two's-complement value.

Switching base

The active base is shown at the top of the screen. Press the base soft keys **DEC**, **HEX**, **OCT**, **BIN** to re-display the current value in another base — the value never changes, only its notation.

Example 1 To view the number 250 in every base

Base-N → 250 → HEX / OCT / BIN

250 in every base

Base	Value
DEC	250
HEX	FA
OCT	372
BIN	11111010

Enter 250 in DEC, then tap each base key. The single quantity 250 is **FA** in hexadecimal, **372** in octal and **11111010** in binary.

Example 2 To convert hexadecimal FF to decimal

Base-N ▸ HEX → FF → DEC

```
Base HEX → DEC
FF
255
```

Type FF while the base is HEX — the hex digits **A–F** are typed with the sin cos tan log ln x^2 keys (A, B, C, D, E, F respectively) — then switch to DEC to read **255**.

Arithmetic in the active base

The four operations **+ – × ÷** (and parentheses) work directly on numbers in the current base; the result is displayed in that base. Only digits the base allows can be entered (0–1 for BIN, 0–7 for OCT, 0–9 and A–F for HEX).

Example 3 To add 1F + 1 in hexadecimal

HEX → 1F + 1 → EXE

```
Base HEX
1F + 1
20
```

In HEX, 1F (= 31) plus 1 is **20** (= 32). Division discards any remainder, since Base-N is integer-only.

Mixing bases with per-value prefixes

From the CATALOG you can tag an individual number with the base to read it in, whatever the active base: **d→** (decimal), **h→** (hexadecimal), **b→** (binary) and **o→** (octal). This lets you combine values written in different bases in one calculation.

Example 4 To add h→FF and 1 while the display base is DEC

DEC CATALOG h→ FF + 1 → EXE

```
Base DEC
h→FF + 1
256
```

The prefix **h→** reads FF as hexadecimal (255) even though the display base is decimal, so h→FF + 1 = **256**. Use d→, b→ or o→ the same way to drop in a decimal, binary or octal literal.

Bitwise logic operators

Press CATALOG in Base-N to insert a logic operator: **and**, **or**, **xor**, **xnor**, **not** and **Neg** (two's-complement negate). These operate on the binary bit patterns.

Example 5 1100 and 1010 (in binary)

BIN → 1100 and 1010 → EXE

```
Base BIN
1100 and 1010
1000
```

and keeps bits that are set in BOTH operands: only the leading bit is set in both, so the result is 1000.

Example 6 1100 or 1010 (in binary)

BIN → 1100 or 1010 → EXE

```
Base BIN
1100 or 1010
1110
```

or keeps bits set in EITHER operand, giving 1110.

Example 7 1100 xor 1010 (in binary)

BIN → 1100 xor 1010 → EXE

```
Base BIN
1100 xor 1010
110
```

xor keeps bits set in EXACTLY ONE operand: 1100 xor 1010 = 0110 (shown as 110).

Note

xnor is the complement of xor (matching bits give 1); **not** inverts every bit of the 32-bit pattern; **Neg** forms the two's-complement negative — Neg 5 = –5. Because not and Neg flip all 32 bits, their results show as a long high-bit pattern in HEX/BIN, exactly like the negative value below.

Negative values

Decimal shows a signed value; the other bases show the 32-bit two's-complement bit pattern, so a negative number prints as its unsigned pattern.

Example 8 To view –5 in two's-complement form

Base-N → –5 → HEX / BIN

–5 in two's complement

Base	Value
DEC	–5
HEX	FFFFFFFB
BIN	…11111011

In DEC, –5 keeps its sign. In HEX it is **FFFFFFFB**, and in BIN the full 32-bit pattern ends in …11111011.

Important!

Base-N is integer-only: fractions, decimals and roots are not available, and a result outside the signed 32-bit range (–2 147 483 648 … 2 147 483 647) is a Math ERROR.

Matrix App Reference

Matrix

Matrix arithmetic up to 4×4

The **Matrix** app performs calculations with matrices of up to four rows and four columns (and rectangular m×n matrices). You store values in the matrix memories **MatA**, **MatB**, **MatC** and **MatD**, then build an expression from them. The result of every matrix calculation is retained in a special memory called **MatAns**, which you can feed straight into the next calculation.

Creating and entering a matrix

When you open an operation, the calculator first asks for the **dimensions** (rows × columns), then shows an empty grid. Fill it cell by cell, pressing EXE after each value to advance to the next cell. This grid holds MatA = $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$:

MatA 2×2

	c1	c2
r1	2	1
r2	1	1

Choose 2 rows and 2 columns, then key in the four values:
2 › EXE › 1 › EXE › 1 › EXE › 1 › EXE

Important!
A matrix memory keeps its contents until you overwrite it or clear the app, so you can reuse MatA in several calculations without re-entering it.

The examples below all use these four matrices:

MatA = $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ **MatB** = $\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$ **MatC** = $\begin{bmatrix} 1 & 0 & -1 \\ 0 & -1 & 1 \end{bmatrix}$ **MatD** = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Addition and Subtraction

Example 1 To add two matrices (MatA + MatB)

Matrix › MatA + MatB → enter MatA, MatB → EXE

MatAns=
$$\begin{bmatrix} 4 & 4 \\ 3 & 2 \end{bmatrix}$$

Corresponding entries are added: the (1,1) entries 2 + 2 = 4, the (1,2) entries 1 + 3 = 4, and so on, giving $\begin{bmatrix} 4 & 4 \\ 3 & 2 \end{bmatrix}$.

Example 2 To subtract two matrices (MatA – MatB)

Matrix ▸ **MatA – MatB** → enter MatA, MatB → **EXE**

```
MatAns=
```

$$\begin{bmatrix} 0 & -2 \\ -1 & 0 \end{bmatrix}$$

Subtraction is also entry by entry: 2 – 2 = 0, 1 – 3 = –2, and so on.

> **Note**
> The two matrices must have the same dimensions in order to be added or subtracted. An error occurs if you try to add or subtract matrices of different dimensions.

Multiplication

Example 3 To multiply two matrices (MatA × MatB)

Matrix ▸ **MatA × MatB** → enter MatA, MatB → **EXE**

```
MatAns=
```

$$\begin{bmatrix} 6 & 7 \\ 4 & 4 \end{bmatrix}$$

Each result entry is a **row × column** dot product. The (1,1) entry is (row 1 of MatA)·(column 1 of MatB) = 2·2 + 1·2 = 6; the (1,2) entry is 2·3 + 1·1 = 7.

Example 4 To multiply matrices of different sizes (MatA × MatC)

Matrix ▸ **MatA × MatB** → MatA (2×2), MatC (2×3) → **EXE**

```
MatAns=
```

$$\begin{bmatrix} 2 & -1 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

A 2×2 times a 2×3 gives a 2×3 result, because MatA has 2 columns and MatC has 2 rows.

> **Important!**
> Matrix multiplication is only possible when the number of columns of the left matrix equals the number of rows of the right matrix. Note that MatA × MatB and MatB × MatA are generally **not** the same.

Powers — Square and Cube

A square matrix can be raised to a power. **MatA²** means MatA × MatA, and **MatA³** means MatA × MatA × MatA.

Example 5 To square and cube MatA ($MatA^2$, $MatA^3$)

Matrix ▸ $MatA^2$ / $MatA^3$ → EXE

```
MatAns=
```
$$\begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}$$

$MatA^2 = \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}$. Cubing goes one step further: $MatA^3 = MatA^2 \times MatA = \begin{bmatrix} 13 & 8 \\ 8 & 5 \end{bmatrix}$.

Inverse

The inverse $MatA^{-1}$ is the matrix that satisfies $MatA \times MatA^{-1} = I$ (the identity). It exists only for a square matrix whose determinant is non-zero.

$$\begin{bmatrix} a_{11} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{a_{11}} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{\begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}}{a_{11}a_{22} - a_{12}a_{21}}$$

Example 6 To invert MatA ($MatA^{-1}$)

Matrix ▸ MatA det / inv / trans ▸ Inverse Matrix → EXE

```
MatAns=
```
$$\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

Because det(MatA) = 1, the inverse is $\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$. You can check it: $MatA \times MatA^{-1}$ returns the identity matrix.

Note

- Only square matrices (same number of rows and columns) can be inverted. Trying to invert a matrix that is not square produces an error.
- A matrix with a determinant of zero cannot be inverted — for example MatD above has det 0, so $MatD^{-1}$ is an error.
- Calculation precision is affected for matrices whose determinant is near zero.

Determinant

The determinant is a single number describing a square matrix. For 1×1, 2×2 and 3×3 matrices it is computed as:

$$\det \begin{bmatrix} a_{11} \end{bmatrix} = a_{11}$$

$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$\det\begin{bmatrix} a_{11} \end{bmatrix} = a_{11}$$

$$\det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$\det\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32}$$

Example 7 To obtain the determinant of MatA (Det(MatA))

Matrix ▸ **MatA det / inv / trans** ▸ **Determinant** → **EXE**

```
MatAns=
        [1]
```

det(MatA) = $a_{11}a_{22} - a_{12}a_{21} = 2\cdot1 - 1\cdot1 = \mathbf{1}$. A determinant is a scalar, so the result is a single value.

Note

Determinants can be obtained only for square matrices. Trying to obtain a determinant for a non-square matrix produces an error. The determinant of MatD (above) is 0, which is why it has no inverse.

Transpose

The transpose M^T turns rows into columns. It works for any matrix, including rectangular ones.

Example 8 To transpose MatC (Trn(MatC))

Matrix ▸ **MatA det / inv / trans** ▸ **Transpose** → MatC → **EXE**

MatAns=

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

The 2×3 matrix MatC becomes a 3×2 matrix — row 1 (1, 0, –1) becomes column 1, and row 2 (0, –1, 1) becomes column 2.

Identity Matrix

Identity(n) creates the n×n identity matrix — 1s on the main diagonal and 0s everywhere else. Multiplying any matrix by the identity leaves it unchanged. The size **n** may be 1 to 10; a value outside that range gives an **Argument ERROR**.

Example 9 To create the 3×3 identity matrix (Identity(3))

Matrix ▸ Identity(n) → 3 → EXE

```
MatAns=
```

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

Enter the size n = 3. Larger results that overflow the screen can be scrolled with the ▲▼◀▶ keys.

Reusing a result — MatAns

Every matrix result is held in **MatAns**. On a result screen press EXE to carry it straight into the next calculation as operand A, without re-entering it.

Example 10 To verify an inverse — MatA × MatAns after inverting MatA

$MatA^{-1}$ → EXE → MatA × MatAns → EXE

```
MatAns=
```

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

Invert MatA first (MatAns becomes $\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$). Press EXE to reuse it, pick **MatA × MatAns**, and the product is the identity $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ — confirming the inverse is correct.

Note

MatAns always holds the most recent matrix result, so you can chain calculations. A scalar result such as a determinant is kept in the ordinary Ans instead.

Copying a result into MatA–MatD

MatAns is overwritten by the next matrix calculation. To **keep** a result, copy it into one of the named matrices. On a matrix result screen press VARIABLE and choose the destination — **MatA**, **MatB**, **MatC** or **MatD**; the whole matrix (its dimensions and every element) is stored there.

Example 11 To save the inverse of MatA into MatB

MatA⁻¹ → EXE → VARIABLE MatB

```
MatAns=
```

$$\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

With the inverse $\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$ on screen, VARIABLE ▸ **MatB** copies it into MatB. It now survives further calculations, so you can build an expression such as MatA × MatB later without recomputing the inverse.

Absolute value of a matrix

Applying **Abs** to a matrix takes the absolute value of **every element**, returning a matrix of the same size. Insert Abs from the FUNCTION menu (or the CATALOG) and put the matrix inside.

Example 12 To take the element-wise absolute value of a matrix

FUNCTION Abs MatA → EXE

```
MatAns=
```

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

With MatA = $\begin{bmatrix} -1 & 2 \\ -3 & 4 \end{bmatrix}$, Abs(MatA) returns $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ — each entry made non-negative. (For a *vector*, Abs instead gives its magnitude |v|.)

Vector App Reference

Vector

2D & 3D vector operations

The **Vector** app stores two vectors, **VctA** and **VctB**, in 2 or 3 dimensions and computes their sum, difference, dot product, cross product, magnitudes, the angle between them and their unit vectors.

Choosing dimensions

```
Vector Dimension
1 2D vectors (x, y)
2 3D vectors (x, y, z)
```

Pick 2D or 3D, then enter the components of VctA and VctB. The cross product is always a vector: for 2D vectors it is (0, 0, z) — the calculator treats them as lying in the z = 0 plane — and for 3D it is the usual perpendicular vector.

$$A \cdot B = AxBx + AyBy + AzBz \qquad |A| = \sqrt{(Ax^2 + Ay^2 + Az^2)} \qquad \cos\theta = \frac{A \cdot B}{|A|\,|B|}$$

Example 1 3D vectors VctA = (1, 2, 3), VctB = (4, 5, 6)

Vector › 3D vectors → enter A, B → FUNCTION

```
VECTOR RESULT (DEG)
VctA · VctB                32
VctA × VctB       (−3, 6, −3)
|VctA|           3.7416573868
Angle           12.9331544919
```

The **dot product** is 1·4 + 2·5 + 3·6 = 32; the **cross product** (–3, 6, –3) is perpendicular to both; **|VctA|** = √14 ≈ 3.742; and the **angle** is about 12.93° (Degree mode).

Key operation:

1 › EXE › 2 › EXE › 3 › EXE › 4 › EXE › 5 › EXE › 6 › EXE › FUNCTION

Example 2 2D vectors VctA = (3, 4), VctB = (1, 0)

Vector › 2D vectors → enter A, B → FUNCTION

```
2D RESULT
VctA · VctB                 3
VctA × VctB        (0, 0, −4)
|VctA|                      5
Angle           53.1301023542
```

In 2D the cross product is the vector **(0, 0, –4)**: the z-component –4 is the signed area of the parallelogram (3·0 – 4·1). |VctA| = √(3² + 4²) = 5, and the angle between the vectors is 53.13°.

Example 3 Sum and difference — VctA + VctB and VctA – VctB (3D vectors)

FUNCTION ▸ VctA + VctB / VctA – VctB

```
SUM & DIFFERENCE
VctA + VctB            (5, 7, 9)
VctA − VctB         (−3, −3, −3)
```

Vectors add and subtract component by component. Using VctA = (1, 2, 3) and VctB = (4, 5, 6): VctA + VctB = **(5, 7, 9)** and VctA – VctB = **(–3, –3, –3)**. Both operands must have the same dimension.

Example 4 The unit vector Â of VctA = (3, 4)

FUNCTION ▸ Unit Vector Â

```
UNIT VECTOR
|VctA|                         5
Â (unit)              (0.6, 0.8)
```

The unit vector Â = A / |A| = (3/5, 4/5) = (0.6, 0.8) has length 1 and points the same way as A.

Reusing a result — VctAns

Every vector result is kept in **VctAns**, exactly like **MatAns** for matrices. Insert it from the **CATALOG** (in the Calculate app) to carry a vector result straight into the next expression.

Example 5 To take the magnitude of a sum — |VctAns| after VctA + VctB

FUNCTION ▸ VctA + VctB → **EXE** → Calculate **CATALOG** VctAns

```
VCTANS CHAINING
VctA + VctB → VctAns    (5, 7, 9)
|VctAns|              12.449899598
```

Using VctA = (1, 2, 3), VctB = (4, 5, 6): the sum (5, 7, 9) is held in VctAns. Reuse it in the Calculate app — for example |VctAns| = $\sqrt{(5^2 + 7^2 + 9^2)} = \sqrt{155} \approx$ **12.45** — without re-entering the vector.

Note

The angle uses the current Angle Unit — in Radian mode the same vectors give the angle in radians. VctAns always holds the most recent vector result, and the Calculate app's CATALOG provides VctA–VctD and VctAns so a vector can be carried into a free expression.

Note

A **scalar** vector result (such as a dot product) can be stored into a memory: on the result screen press **VARIABLE** and pick A–F, x, y or z. See “Storing a result from another app” in the Calculate chapter.

Ratio App Reference

Ratio

Solving proportions

The **Ratio** app finds the missing term X in a proportion A : B = C : D. Choose which position is unknown, enter the three known values, and the calculator solves for X by cross-multiplication.

Choosing the form

```
Ratio Form
1 A : B = X : D
2 A : B = C : X
```

Form 1 solves for the third term; form 2 solves for the fourth. Both use the rule that equal ratios cross-multiply: A · D = B · C.

Example 1 To solve 3 : 4 = X : 8

Ratio ▸ A : B = X : D → 3, 4, 8 → FUNCTION

```
A : B = X : D
A                3
B                4
D                8
X                6
```

X = (A · D) / B = (3 · 8) / 4 = **6**. This is just the proportion 3⁄4 = X⁄8, so 4X = 24.

Key operation:

3 › EXE › 4 › EXE › 8 › EXE › FUNCTION

Example 2 To solve 3 : 4 = 6 : X

Ratio ▸ A : B = C : X → 3, 4, 6 → FUNCTION

```
A : B = C : X
A                3
B                4
C                6
X                8
```

X = (B · C) / A = (4 · 6) / 3 = **8**.

Example 3 A zero divisor is rejected (3 : 0 = X : 8)

Ratio ▸ A : B = X : D → 3, 0, 8 → FUNCTION

```
A : B = X : D
A                3
B                0
D                8
        Math ERROR
```

Solving needs a division by B (form 1) or A (form 2), so a **0** in that position makes the proportion undefined and returns **Math ERROR**. Re-enter a non-zero value to continue.

Note

Each ratio box opens pre-filled with **1** (A = 1, B = 1, D = 1 or C = 1), so you only overwrite the terms you know. Two ratios are equal exactly when their cross-products match (A · D = B · C).

Spreadsheet App Reference

Spreadsheet

A calculator spreadsheet

The **Spreadsheet** app is a grid of **5 columns (A–E) × 45 rows**. Each cell holds either a number or a formula that begins with **=** and may reference other cells (A1, B2, …). Formulas recalculate automatically as you edit.

Entering constants

Move the cursor with the arrow keys and type a number to enter a constant into a cell. Press **EXE** to confirm and drop to the next row. The examples in this chapter build on column A holding the values 1 to 5.

Example 1 To enter the values 1–5 into cells A1 to A5

Spreadsheet → point to A1 → **1** **EXE** **2** **EXE** …

Spreadsheet

	A
1	1
2	2
3	3
4	4
5	5

Each value is typed and confirmed with **EXE**, which moves down to the next cell. The reference of a cell is its column letter and row number, so the value 3 sits in **A3**.

Entering formulas

Start a cell with **=** to make it a formula. A formula may contain numbers, operators and **cell references** (A1, B2, …). A reference always tracks the current value of that cell, so the formula recalculates automatically whenever the referenced cell changes.

Example 2 To put =A1×2 in cell B1

point to B1 → **=** A1 **×** **2** **EXE**

Spreadsheet

	A	B
1	1	2
2	2	
3	3	
4	4	
5	5	

B1 shows its computed value **2** (= 1 × 2). If you later change A1, B1 updates by itself. Use **Show Cell** (TOOLS) to view the underlying formula =A1×2 instead of the value.

Note

Enter the **column letters** A–E of a reference with the **sin** **cos** **tan** **log** **ln** keys (A, B, C, D, E respectively) and the row number with the number keys — so **A1** is **sin** **1** and **B2** is **cos** **2**.

Relative and absolute references

When a formula is copied or filled to another cell, its references shift with it — these are **relative** references. A **$** locks part of a reference so it does **not** shift: **A1** is fully locked, **A$1** locks only the row and **$A1** only the column.

Example 3 To fill =A1×2 down column B (relative reference)

B1 → TOOLS ▸ Fill Formula → range B1:B5 → EXE

Spreadsheet

	A	B
1	1	2
2	2	4
3	3	6
4	4	8
5	5	10

Filling adjusts the relative reference for each row: B2 becomes =A2×2, B3 becomes =A3×2, and so on — giving 2, 4, 6, 8, 10.

Example 4 To fill =A1×A5 down column C (absolute reference)

C1 → TOOLS ▸ Fill Formula → range C1:C5 → EXE

Spreadsheet

	A	C
1	1	5
2	2	10
3	3	15
4	4	20
5	5	25

The A5 part is locked to cell A5 (= 5) in every row, while A1 shifts to A2, A3, … So the column becomes A × 5 = 5, 10, 15, 20, 25.

Range commands — Sum, Min, Max, Mean

The commands **Sum(**, **Min(**, **Max(** and **Mean(** operate over a block of cells written as a range such as A1:A5. Insert them from the **SHEET** commands in the CATALOG.

Example 5 To total column A with =Sum(A1:A5)

CATALOG ▸ SHEET ▸ Sum(→ A1:A5 → EXE

```
Sheet
=Sum(A1:A5)
15
```

Sum adds every value in the range: 1 + 2 + 3 + 4 + 5 = **15**.

Example 6 To find the mean, maximum and minimum of A1:A5

CATALOG ▸ SHEET ▸ Mean(/ Max(/ Min(→ A1:A5 → EXE

```
RANGE COMMANDS
=Mean(A1:A5)        3
=Max(A1:A5)         5
=Min(A1:A5)         1
```

Mean returns the average (15 ÷ 5 = 3), while Max and Min return the largest and smallest values in the range.

Batch input — Fill Value

To put the **same constant** into many cells at once, use **Fill Value** instead of typing it row by row.

Example 7 To write the constant 100 into D1:D3 with Fill Value

TOOLS ▸ Fill Value → Value 100, range D1:D3 → EXE

Spreadsheet

	D
1	100
2	100
3	100

Fill Value writes the constant into every cell of the range in one step. Its companion **Fill Formula** (Examples 3–4) does the same for a formula, adjusting the relative references as it goes.

The spreadsheet TOOLS menu

Press TOOLS inside the sheet for editing commands that act on the pointed cell or range:

Command	What it does
Fill Formula	Copies a formula across a range, adjusting relative references (locked $ parts stay put).
Fill Value	Writes the same constant into every cell of a range.
Copy & Paste	Duplicates a cell, adjusting its relative references at the destination.
Cut & Paste	Moves a cell; references are kept and the source is cleared.
Grab	Point to a cell to insert its reference into the formula you are editing.
Show Cell	Toggles the grid between showing computed **values** and the underlying **formulas**.
Auto Calc	Turns automatic recalculation on or off; Recalculate refreshes on demand.
Delete All	Clears the whole spreadsheet.

Important!

The spreadsheet holds up to 5 × 45 = 225 cells. Turning Auto Calc off freezes the displayed values — useful while entering a large sheet — until you Recalculate.

Math Box App Reference

Math Box

Learning-support tools

The **Math Box** app is a set of interactive learning aids: probability experiments and visual number tools. Choose a tool from the menu.

The Math Box menu

Math Box
1 Dice Roll
2 Coin Toss
3 Number Line
4 Circle

Each tool runs an experiment or draws a figure you can explore. Point to a tool and press **OK**.

Example 1 Dice Roll — 2 dice, 20 rolls, tallied by Sum

Math Box ▸ Dice Roll → dice 2, attempts 20 → **EXE** ▸ Frequency

Dice Roll · Sum

Sum	Freq
5	3
6	5
7	6
8	4

Rolls 1–3 dice for a chosen number of attempts and tallies each outcome — a hands-on way to see experimental probability approach the theoretical values. After the run, choose **List** (every roll) or the **Frequency** table shown here. The **Same Result** setting (Off / #1 / #2 / #3) replays a fixed random sequence, so a whole class sees identical rolls (the dice count, attempts and Same Result slot must match on every calculator).

Example 2 Dice Roll — the same 2 dice tallied by Difference

Dice Roll → dice 2, attempts 20 → **EXE** ▸ Frequency ▸ Difference

Dice Roll · Difference

Diff	Freq
0	3
1	6
2	4
3	3
4	2
5	2

With **two** dice a second menu asks whether to tally by **Sum** (2–12) or by **Difference** (0–5) — the absolute gap between the two dice. Difference 1 is the most common, difference 5 the rarest, matching the theoretical shape. (This is one 20-roll run; your frequencies will differ unless Same Result is set.)

Example 3 Coin Toss — 2 coins, 20 tosses

Math Box ▸ Coin Toss → coins 2, attempts 20 → EXE ▸ Frequency

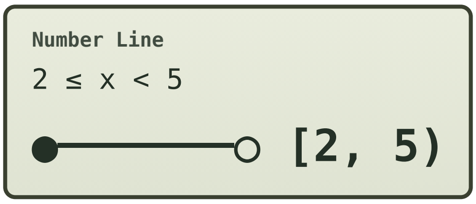
Coin Toss

Heads	Freq
0	5
1	11
2	4

With two coins the middle outcome (one head, one tail) is about twice as likely as two heads or two tails — visible directly in the tally. Coin Toss offers the same List / Frequency choice as the dice.

Example 4 Number Line — draw 2 ≤ x < 5

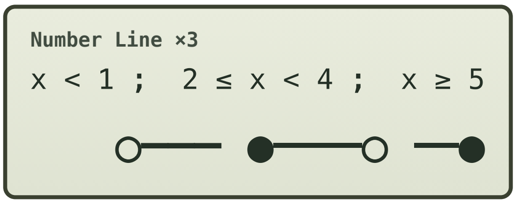
Math Box ▸ Number Line ▸ a ≤ x < b → 2, 5



Draws the solution of a simple inequality: a filled circle for an inclusive bound, an open circle for a strict one. Nine expression types are available, from x < a to a ≤ x ≤ b.

Example 5 Number Line — up to three regions at once

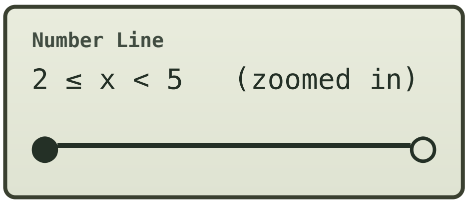
Number Line → slot 1, slot 2, slot 3 → EXE



The Number Line holds up to **three** expression slots and draws them on one axis, so regions can be compared side by side — here x < 1, the band 2 ≤ x < 4, and x ≥ 5. Any of the nine expression types can go in any slot.

Example 6 Number Line — pan and zoom the view-window

on the drawn line: ◀ ▶ pan · ▲ ▼ zoom · DEL reset



The line first auto-fits to show every region with a little margin. On the drawn line you can then adjust the **view-window**: ◀ ▶ pan left / right, ▲ zoom in and ▼ zoom out (about the centre), and DEL snaps back to the automatic fit. Zooming in spreads a tight cluster of bounds apart so they are easier to read; zooming out reveals a region that sits far off to one side.

Example 7 Circle — the Unit Circle at 30°

Math Box ▸ Circle ▸ Unit Circle → θ 30 → EXE

```
CIRCLE · UNIT
θ1                30°
sin               0.5
cos      0.8660254038
tan      0.5773502692
```

The point on the unit circle at 30° is (x, y) = (cos θ, sin θ) = (0.866, 0.5). The read-out lists **sin**, **cos** and **tan** of the angle — a visual link between an angle and its trig ratios. The three Circle types are **Unit Circle**, **Half Circle** and **Clock**.

Example 8 Circle — two angles at once (45° and 135°)

Circle ▸ Unit Circle → θ_1 45, θ_2 135 → EXE

```
CIRCLE · TWO ANGLES
θ1                45°
sin      0.7071067812
cos      0.7071067812
tan                 1
```

Enter a second angle θ_2 to draw two radii together; the selected one is drawn thicker and its sin/cos/tan are shown. Press ▲ ▼ to switch to θ_2 = 135° (sin 0.7071, cos –0.7071, tan –1). **Half Circle** works the same way but restricts angles to 0°–180° (0–π in Radian mode).

Example 9 Circle — the Clock at 3:00

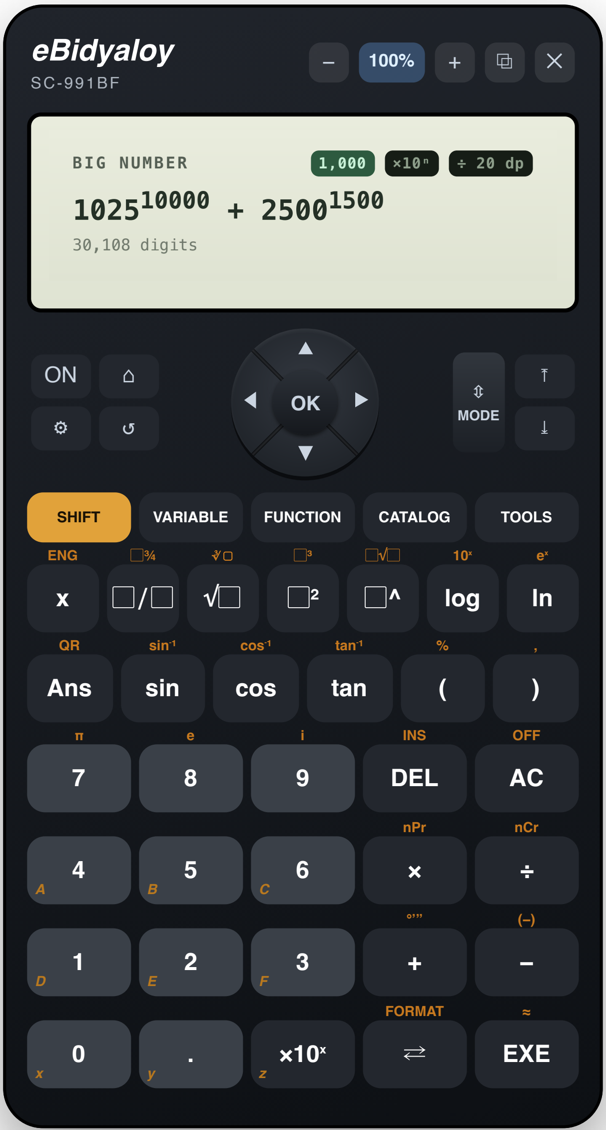
Circle ▸ Clock → hour 3 → EXE

```
CIRCLE · CLOCK
Time              3:00
θ1 (smaller)       90°
θ2 (larger)       270°
```

The Clock shows the two angles between the hour and minute hands: at 3:00 they are **θ_1 = 90°** (the smaller) and **θ_2 = 270°** (the larger, $\theta_1 + \theta_2$ = 360°). Press ▲▼ to advance or rewind the hour one step at a time.

> **Note**
> Math Box tools are for exploration and teaching; they don't feed results into the other apps, but the frequency tables and coordinates can be read off directly.

Big Number App



eBidyaloy · EXTENDED DISPLAY ● LIVE DISPLAY

BIG NUMBER 1,000 ×10ⁿ ÷ 20 dp

$1025^{10000} + 2500^{1500}$

30,108 digits

173,242,290,766,589,092,039,475,342,803,274,971,290,484,19
4,579,052,361,896,224,613,232,857,912,237,233,991,945,983,
968, …

$1025^{10000} + 2500^{1500} =$

30,108 digits

COPY

```
173,242,290,766,589,092,039,475,342,803,274,971,290,484,194,579,052,361,896,224,613,232,85
7,912,237,233,991,945,983,968,360,118,869,390,843,174,584,171,623,979,420,747,431,009,193,6
67,701,011,872,908,579,395,378,858,343,595,117,111,468,506,005,189,905,804,623,450,066,137,
389,129,784,624,618,752,935,587,733,656,589,902,790,034,062,426,305,881,887,637,874,080,44
4,838,074,060,918,718,961,248,001, …
```

The floating calculator (top) and its live extended display with the Big Number explain window (below).

The **Big Number** app does exact arithmetic on **very large whole numbers** — hundreds or thousands of digits — and shows the answer **in full, never in exponential form**. An ordinary calculator would round 1025^{10000} to a short $a \times 10^{n}$ value; Big Number prints every one of its 30,108 digits.

Opening Big Number

HOME ▸ Big Number

Press the ⌂ (HOME) key to open the app list and choose **Big Number**. (Big Number is opened from HOME — not from the blue **MODE** button, which only switches Normal / Step-by-Step in the Calculate app.) The screen shows the **BIG NUMBER** banner, an input line, and the result area.

> **Note**
>
> Results are exact integers. A **division** that does not come out evenly is carried to a set number of decimal places instead (see “Division precision”, below) — every other operation (+ – × and powers) is always exact.

Entering an expression

Integers, + – × ÷, powers, parentheses

Type whole numbers with the digit keys and combine them with:

- **+** **–** **×** **÷** — add, subtract, multiply, divide;
- **□^** — a **power**. The exponent is entered as a raised superscript, so 1025^{10000} appears just as written; press **▶** to leave the exponent and continue;
- **(** **)** — parentheses for grouping.

Press **EXE** to evaluate. Use **DEL** to delete and **AC** to clear the line.

> **Important!**
>
> Big Number takes **whole numbers only** as input — there is no decimal point on entry. A decimal result can still appear (from division); it is shown to the chosen number of places.

Reading the result

Digit count, chips, quotient & remainder

After EXE the result area shows a **digit count** (for example "30,108 digits") above the number itself. Three chips at the top-right of the display control how the answer is shown:

Chip	Meaning
1,000	**Thousands grouping** — inserts a comma every three digits so a long number is readable. On by default. This is display only; **Copy** always copies the plain, ungrouped digits.
$\times 10^n$	**Scientific form** — adds an approximate ≈ $a.bcde \times 10^n$ line beneath the full digits, giving the size of the number at a glance. Off by default.
÷ 20 dp	**Division precision** — how many decimal places a division is carried to. Press ▲ / ▼ to cycle 10 · 20 · 50 · 100.

For a plain whole-number division such as `100000 ÷ 7`, a **quotient and remainder** line (= `14,285 R 5`) is shown beneath the decimal value.

Example 1 Whole-number division — decimal, quotient and remainder

1 0 0 0 0 0 ÷ 7 EXE

```
BIG NUMBER          25 digits
100000 ÷ 7
14,285.71428571428571428571
= 14,285 R 5
```

At the default 20 dp the decimal is carried to twenty places; the reminder line gives the exact integer quotient (14,285) and remainder (5).

The extended display & explain window

Present a big result

Press the ⧉ (cast) button to open the **extended display** — a large mirror of the calculator screen for a class or projector, marked **LIVE DISPLAY**. Below it, the **explain window** restates the expression, the digit count, and the **complete grouped result** in a scrolling panel, with a one-tap **COPY** that copies the raw digits. Both refresh automatically after each EXE (shown in the hero at the top of this chapter).

The FUNCTION menu

Number-theory operations with placeholder boxes

Press the FUNCTION key to insert a function. Each function is inserted as a template with empty ☐ **boxes** for its arguments — fill the first box, press ▶ (or ▼) to move to the next, and press ▶ once more to leave the closing bracket. You never type a comma.

Entry	Meaning
`☐!`	Factorial — n! = 1 × 2 × … × n.
`a mod b`	Remainder of a ÷ b.
`gcd(☐, ☐)`	Greatest common divisor.

`lcm(▢, ▢)`	Least common multiple.
`nCr(▢, ▢)`	Combinations — n choose r.
`nPr(▢, ▢)`	Permutations.
`powmod(▢, ▢, ▢)`	Modular power — a^b mod m, computed without ever building the full power.
`root(▢, ▢)`	Integer n-th root — $\lfloor x^{1/n} \rfloor$.
`isqrt(▢)`	Integer square root — $\lfloor \sqrt{x} \rfloor$.
`digitsum(▢)`	Sum of the decimal digits.
`isprime(▢)`	Primality test — 1 if prime, 0 if not.
`sumeven(▢, ▢)` · `sumodd(▢, ▢)`	Sum of the even / odd whole numbers in the range a…b.
`factorize(▢)`	Prime factorisation, as p^e × q × …
`evens(▢, ▢)` · `odds(▢, ▢)`	List the even / odd whole numbers in the range a…b (comma-separated).
`primes(▢)`	List the first n prime numbers.
`fib(▢)`	List the first n Fibonacci numbers (0, 1, 1, 2, 3, …).
`pi(▢)`	π to n decimal places (up to 50,000).
`hex(▢)` · `bin(▢)` · `oct(▢)`	Convert to base 16 / 2 / 8 (0x… / 0b… / 0o…).

Note

factorize, **hex**, **bin**, **oct**, **evens**, **odds**, **primes**, **fib** and **pi** produce text (a factor list, a based number, or a sequence), not a plain integer, so they can only be used on their own — not inside a larger sum. Their argument may itself be any expression, e.g. `factorize(nCr(20, 10))`. Every other function — including **sumeven** and **sumodd** — combines freely: `gcd(48, 36) + 1` works.

Worked examples

Example 2 Factorial — 100!

1 0 0 ▢! EXE

BIG NUMBER 158 digits

100!

**93,326,215,443,944,152,681,699,238,856,266,70
0,490,715,968,264,381,621,468,592,963,895,21
7,599,9…**

100! has 158 digits — exact, with every digit shown.

Example 3 Exact power — 2 to the 1000

2 □^ 1 0 0 0 EXE

BIG NUMBER 302 digits

2^{1000}

10,715,086,071,862,673,209,484,250,490,600,01
8,105,614,048,117,055,336,074,437,503,883,70
3,510,5…

A 302-digit power, computed exactly by repeated squaring.

Example 4 Combinations — nCr(200, 100)

nCr(□,□) 2 0 0 ▶ 1 0 0 EXE

BIG NUMBER 59 digits

nCr(200, 100)

90,548,514,656,103,281,165,404,177,077,484,16
3,874,504,589,675,413,336,841,320

Fill the first box (200), press ▶, fill the second (100), then EXE.

Example 5 Greatest common divisor & least common multiple

gcd(□,□) 123456789 ▶ 987654321 EXE

BIG NUMBER

gcd(123456789, 987654321)

9

BIG NUMBER 10 digits

lcm(123456, 789012)

8,117,355,456

gcd of the two numbers is 9; the lcm example gives 8,117,355,456.

Example 6 Modular power — powmod(7, 100000, 13)

powmod(□,□,□) 7 ▶ 100000 ▶ 13 EXE

BIG NUMBER

powmod(7, 100000, 13)

9

7 to the 100,000 mod 13 = 9, found without ever forming the 84,510-digit power.

Example 7 Sum of digits & primality

digitsum(□) 2 □^ 1000 EXE

```
BIG NUMBER
digitsum(2^1000)
1,366
```

```
BIG NUMBER
isprime(2147483647)
1 (prime)
```

The digits of 2^{1000} add up to 1,366; 2,147,483,647 ($2^{31}-1$) is prime, so isprime returns 1.

Example 8 Prime factorisation

factorize(□) 600851475143 EXE

```
BIG NUMBER
factorize(600851475143)
71 × 839 × 1471 × 6857
```

The result is a factor list (p^e × q × …), so it stands as the whole expression.

Example 9 Integer roots & base conversion

isqrt(□) 10 □^ 60 EXE

```
BIG NUMBER                          31 digits
isqrt(10^60)
1,000,000,000,000,000,000,000,000,000,000
```

```
BIG NUMBER
hex(48879)
0xBEEF
```

isqrt of 10^{60} is 10^{30}; hex(48879) converts to base 16 (0xBEEF). bin(…) and oct(…) work the same way.

Ranges, sequences & π

evens · odds · sumeven · sumodd · primes · fib · pi

Big Number can also list or add up a whole run of numbers and print π to any length. The **list** functions (`evens`, `odds`, `primes`, `fib`) return a comma-separated sequence; `sumeven` and `sumodd` return a single total; and `pi(n)` prints π to *n* decimal places. The list functions are capped so a huge request is refused rather than left to fill the screen.

Example 10 Even & odd numbers in a range

evens(□,□) 1 ▶ 20 EXE

```
BIG NUMBER
evens(1, 20)
2, 4, 6, 8, 10, 12, 14, 16, 18, 20
```

```
BIG NUMBER
odds(1, 20)
1, 3, 5, 7, 9, 11, 13, 15, 17, 19
```

Fill the low end (1), press ▶, fill the high end (20), then EXE. Up to 10,000 terms are listed.

Example 11 Sum of the evens or odds in a range

sumeven(□,□) 1 ▶ 100 EXE

```
BIG NUMBER
sumeven(1, 100)
2,550
```

```
BIG NUMBER
sumodd(1, 100)
2,500
```

sumeven and sumodd add the even (or odd) whole numbers in the range. They return a plain integer, so they combine freely: sumeven(1, 100) + 1 works.

Example 12 First n primes & Fibonacci numbers

primes(□) 10 EXE

```
BIG NUMBER
primes(10)
2, 3, 5, 7, 11, 13, 17, 19, 23, 29
```

```
BIG NUMBER
fib(15)
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,
233, 377
```

primes(n) lists the first n prime numbers; fib(n) lists the first n Fibonacci numbers, starting 0, 1.

Example 13 π to a chosen number of decimal places

pi(□) 50 EXE

```
BIG NUMBER
pi(50)
3.141592653589793238462643383279502884197169З
9937510
```

pi(n) gives π to n decimal places — up to 50,000 — computed with the fast Chudnovsky series.

Really big numbers

Thousands of digits, in full

Big Number handles results up to **50,000 digits**. Powers are guarded so an over-large one is refused rather than left to hang the screen. Two examples that would overflow any ordinary calculator:

Example 14 $1025^{10000} + 2500^{1500}$

1 0 2 5 □^ 10000 ▶ + 2 5 0 0 □^ 1500 EXE

BIG NUMBER 30,108 digits

$1025^{10000} + 2500^{1500}$

173,242,290,766,589,092,039,475,342,803,274,971,290,484,194,579,052,361,896,224,613,232,857,912,…

30,108 digits — with $\times 10^n$ turned on, the scientific-form line summarises the size.

Example 15 $1025^{10000} + 2500^{10000}$

1 0 2 5 □^ 10000 ▶ + 2 5 0 0 □^ 10000 EXE

BIG NUMBER 33,980 digits

$1025^{10000} + 2500^{10000}$

25,123,880,576,987,445,851,801,350,421,336,100,603,336,315,995,959,360,323,468,077,535,592,121,9…

33,980 digits — the full number scrolls in the result area and the explain window.

Turn **$\times 10^n$** on to read the magnitude at a glance, keep **1,000** on to read the digits in groups, and use **COPY** to paste the exact number elsewhere — the copy is always the plain, ungrouped value.

Constants & Unit Conversions

Physical Constants

CATALOG → CONST ▸

The **CONST ▸** sub-menu inserts any of these 47 scientific constants (CODATA-based values) into a calculation. Point to a category, then to a constant, and press **OK** to insert its symbol and value. The six categories and their constants are:

Universal

Symbol	Constant	Value	Unit
h	Planck constant	$6.62607015 \times 10^{-34}$	J·s
ħ	reduced Planck constant (h/2π)	$1.054571817 \times 10^{-34}$	J·s
c	speed of light in vacuum	2.99792458×10^{8}	m/s
ε_0	electric constant	$8.8541878188 \times 10^{-12}$	F/m
μ_0	magnetic constant	$1.25663706127 \times 10^{-6}$	N/A^2
Z_0	vacuum impedance	376.730313412	Ω
G	gravitational constant	6.6743×10^{-11}	$m^3 \cdot kg^{-1} \cdot s^{-2}$
lP	Planck length	1.616255×10^{-35}	m
tP	Planck time	5.391247×10^{-44}	s

Electromagnetic

Symbol	Constant	Value	Unit
μN	nuclear magneton	$5.0507837393 \times 10^{-27}$	J/T
μB	Bohr magneton	$9.2740100657 \times 10^{-24}$	J/T
e	elementary charge	$1.602176634 \times 10^{-19}$	C
Φ_0	magnetic flux quantum	$2.067833848 \times 10^{-15}$	Wb
G_0	conductance quantum	$7.748091729 \times 10^{-5}$	S
KJ	Josephson constant	$4.835978484 \times 10^{14}$	Hz/V
RK	von Klitzing constant	25812.80745	Ω

Atomic & Nuclear

Symbol	Constant	Value	Unit
mp	proton mass	$1.67262192595 \times 10^{-27}$	kg
mn	neutron mass	$1.67492750056 \times 10^{-27}$	kg
me	electron mass	$9.1093837139 \times 10^{-31}$	kg
mμ	muon mass	$1.883531627 \times 10^{-28}$	kg

a_0	Bohr radius	$5.29177210544 \times 10^{-11}$	m
α	fine-structure constant	0.0072973525643	—
re	classical electron radius	$2.8179403205 \times 10^{-15}$	m
λC	Compton wavelength	$2.42631023538 \times 10^{-12}$	m
γp	proton gyromagnetic ratio	$2.6752218708 \times 10^{8}$	$s^{-1}·T^{-1}$
λCp	proton Compton wavelength	$1.3214098536 \times 10^{-15}$	m
λCn	neutron Compton wavelength	$1.31959090382 \times 10^{-15}$	m
R∞	Rydberg constant	$1.09737315682 \times 10^{7}$	m^{-1}
μp	proton magnetic moment	$1.41060679545 \times 10^{-26}$	J/T
μe	electron magnetic moment	$-9.2847646917 \times 10^{-24}$	J/T
μn	neutron magnetic moment	$-9.6623653 \times 10^{-27}$	J/T
μμ	muon magnetic moment	$-4.4904483 \times 10^{-26}$	J/T
mτ	tau mass	3.16754×10^{-27}	kg

Physico-Chem

Symbol	Constant	Value	Unit
mu	atomic mass constant	$1.66053906892 \times 10^{-27}$	kg
F	Faraday constant	96485.33212	C/mol
NA	Avogadro constant	$6.02214076 \times 10^{23}$	mol^{-1}
k	Boltzmann constant	1.380649×10^{-23}	J/K
Vm	molar volume of ideal gas	0.02271095464	m^3/mol
R	molar gas constant	8.314462618	$J·mol^{-1}·K^{-1}$
c_1	first radiation constant	$3.741771852 \times 10^{-16}$	$W·m^2$
c_2	second radiation constant	0.01438776877	m·K
σ	Stefan-Boltzmann constant	$5.670374419 \times 10^{-8}$	$W·m^{-2}·K^{-4}$

Adopted Values

Symbol	Constant	Value	Unit
gn	std acceleration of gravity	9.80665	m/s^2
atm	standard atmosphere	101325	Pa
RK-90	conventional von Klitzing	25812.807	Ω
KJ-90	conventional Josephson	4.835979×10^{14}	Hz/V

Other

Symbol	Constant	Value	Unit
t	Celsius temperature (0 °C)	273.15	K

Note

Values follow the 2022 CODATA recommended set. A constant is inserted as its stored value; combine it with other terms just like any number.

Unit Conversions

CATALOG → CONV ▸

The **CONV ▸** sub-menu converts the value currently on the display. First compute or type a value, then choose a category and a conversion; the display is replaced by the converted value. There are 40 conversions in nine categories:

Length

Conversion	Factor
in ▶ cm	× 2.54
cm ▶ in	÷ 2.54
ft ▶ m	× 0.3048
m ▶ ft	÷ 0.3048
yd ▶ m	× 0.9144
m ▶ yd	÷ 0.9144
mile ▶ km	× 1.609344
km ▶ mile	÷ 1.609344
n mile ▶ m	× 1852
m ▶ n mile	÷ 1852
pc ▶ km	× 3.0856776×10^{13}
km ▶ pc	÷ 3.0856776×10^{13}

Area

Conversion	Factor
acre ▶ m^2	× 4046.856422
m^2 ▶ acre	÷ 4046.856422

Volume

Conversion	Factor
gal(US) ▶ L	× 3.785411784
L ▶ gal(US)	÷ 3.785411784

gal(UK) ▶ L	× 4.54609
L ▶ gal(UK)	÷ 4.54609

Mass

Conversion	Factor
oz ▶ g	× 28.349523125
g ▶ oz	÷ 28.349523125
lb ▶ kg	× 0.45359237
kg ▶ lb	÷ 0.45359237

Velocity

Conversion	Factor
km/h ▶ m/s	÷ 3.6
m/s ▶ km/h	× 3.6

Pressure

Conversion	Factor
atm ▶ Pa	× 101325
Pa ▶ atm	÷ 101325
mmHg ▶ Pa	× 133.322387415
Pa ▶ mmHg	÷ 133.322387415
kgf/cm^2 ▶ Pa	× 98066.5
Pa ▶ kgf/cm^2	÷ 98066.5
lbf/in^2 ▶ kPa	× 6.894757293
kPa ▶ lbf/in^2	÷ 6.894757293

Energy

Conversion	Factor
kgf·m ▶ J	× 9.80665
J ▶ kgf·m	÷ 9.80665
J ▶ cal$_{15}$	÷ 4.1858
cal$_{15}$ ▶ J	× 4.1858

Power

Conversion	Factor
hp ▶ kW	× 0.745699872
kW ▶ hp	÷ 0.745699872

Temperature

Conversion	Factor
°F ▶ °C	(x – 32) × 5⁄9
°C ▶ °F	x × 9⁄5 + 32

Note

Factors follow NIST Special Publication 811. Each conversion has a matching reverse conversion (for example in ▶ cm and cm ▶ in), so you can convert in either direction.

Technical Reference

Calculation Priority Sequence

Order of operations

The calculator evaluates an expression according to a fixed priority sequence. Basically, calculations run from left to right, expressions inside parentheses have the highest priority, and each command has the priority shown below.

Priority	Commands
1	Parenthetical expressions
2	Functions that take parentheses — sin(, cos(, tan(, ln(, log(, √, ∛, log▢(, Abs(, GCD(, LCM(, d/dx, ∫dx, Σ, Π, Pol(, Rec(and similar
3	Functions that come after a value — x^2, x^3, x^{-1}, x!, %, angle units ° ʳ ᵍ — and powers ($x^{\square}$) and roots ($\sqrt[\square]{\square}$)
4	Negative sign (–) and Base-N prefixes (d, h, b, o)
5	Permutation (nPr), combination (nCr), complex polar symbol (∠)
6	Implicit multiplication — an omitted × before a value, constant or bracket (2π, 3(1+2))
7	Multiplication (×), division (÷), dot product (·)
8	Addition (+), subtraction (–)
9	and (logical operator, Base-N)
10	or, xor, xnor (logical operators, Base-N)

Important!

Implicit multiplication (priority 6) binds **tighter** than explicit × and ÷ (priority 7). This is why **6 ÷ 2(1 + 2) = 1** — the calculator reads it as 6 ÷ (2 × (1 + 2)) — and **6 ÷ 2π = 0.9549…** reads as 6 ÷ (2π).

Precautions when a calculation contains negative values

Because a postfix function such as x^2 (priority 3) binds tighter than the negative sign (priority 4), a leading minus applies to the whole power. To square a negative value you must enclose it in parentheses.

-2^2 = -4

Math DEG
-2^2
−4

reads $-(2^2)$: the square is taken first, then negated

$(-2)^2$ = 4

Math DEG
$(-2)^2$
4

the parentheses square the whole value −2

Calculation Ranges, Digits and Precision

How the emulator computes

The SC-991BF emulator performs every calculation in **IEEE-754 double precision** — the same arithmetic used throughout modern computing — and then formats the answer the way a scientific calculator would.

Property	Value
Internal precision	Double precision — about 15–16 significant digits
Displayed digits (Norm)	Decimal results shown to up to 6 decimal places (trailing zeros removed); integer part shown in full. Use Fix 7–9 / Sci for more
Switch to scientific form	When $|x| \geq 1\times10^{10}$ or $|x| < 1\times10^{-9}$ (Norm 2, default); $< 1\times10^{-2}$ in Norm 1
Fix setting	Fixed number of decimal places, 0 to 9
Sci setting	Fixed number of significant figures

Note

Errors are cumulative across consecutive calculations, and tend to be larger near a function's singular points and inflection points. Functions that need repeated internal calculation — x^y, $\sqrt[x]{y}$, x!, nPr, nCr, numerical d/dx and ∫dx — can accumulate error with each step.

Function input ranges

The following table lists the practical input range of the main functions. Values outside a range give a **Math ERROR**.

Function	Input range
sin x, cos x, tan x	Degree: $|x| < 9\times10^{9}$ · Radian: $|x| < 157079632.7$ · (tan x undefined at odd multiples of 90°)
$\sin^{-1}x$, $\cos^{-1}x$	$0 \leq |x| \leq 1$
$\tan^{-1}x$	$|x| \leq 9.999999999\times10^{99}$
sinh x, cosh x	$|x| \leq 230.2585092$
$\sinh^{-1}x$	$|x| \leq 4.999999999\times10^{99}$
$\cosh^{-1}x$	$1 \leq x \leq 4.999999999\times10^{99}$
tanh x	$|x| \leq 9.999999999\times10^{99}$
$\tanh^{-1}x$	$|x| \leq 9.999999999\times10^{-1}$
log x, ln x	$0 < x \leq 9.999999999\times10^{99}$
10^x	$-9.999999999\times10^{99} \leq x \leq 99.99999999$
e^x	$-9.999999999\times10^{99} \leq x \leq 230.2585092$
$\sqrt{x}$	$0 \leq x < 1\times10^{100}$
x^2	$|x| < 1\times10^{50}$
x^{-1}	$|x| < 1\times10^{100}$; $x \neq 0$
x!	$0 \leq x \leq 69$ (x is an integer)
nPr, nCr	$0 \leq r \leq n$; $n < 1\times10^{10}$ (n, r integers)
Pol(x, y)	$|x|, |y| \leq 9.999999999\times10^{99}$
Rec(r, θ)	$0 \leq r \leq 9.999999999\times10^{99}$; θ same as sin x

RanInt#(a, b)	a < b; |a|, |b| < 1×10^{10}
GCD(a, b), LCM(a, b)	integers, |a|, |b| < 1×10^{10}

Note

The Base-N app works only with integers in the signed 32-bit range –2 147 483 648 to 2 147 483 647; a result outside this range is a Math ERROR.

Important!

Tangent singularities. tan x is undefined at odd multiples of 90° (90°, 270°, …; π/2, 3π/2, … in Radian mode), where the curve goes to infinity. Entering tan 90° gives a **Math ERROR** rather than a huge finite number. The trigonometric functions also reject arguments beyond about |x| = 9×10^9 degrees, where rounding would make the result meaningless — this too is reported as a Math ERROR.

Error Messages

Causes and remedies

When a calculation cannot be completed, the calculator shows an error message instead of a result and returns you to the expression with the **cursor already placed on the item that caused the problem** — so you can correct it immediately without searching. (Press ◀ or ▶ to move on from there.) The common messages are:

Message	Cause	Remedy
Syntax ERROR	The expression is not written correctly — for example mismatched parentheses, or an operator with a missing operand.	Check the input at the cursor and correct the format.
Math ERROR	The calculation is mathematically undefined or out of range — division by zero, √ of a negative in real mode, or a value outside a function's input range (see the table above).	Check the values. For a square root or power that gives a complex result, use the **Complex** app.
Dimension ERROR	A matrix or vector operation was attempted on incompatible sizes — for example adding matrices of different dimensions, or multiplying when the columns of the first do not equal the rows of the second.	Re-enter the matrices or vectors with compatible dimensions.
Square only	A determinant or inverse was requested for a matrix that is not square.	Use a square matrix (equal rows and columns).
Argument ERROR	A command was given an argument outside its allowed set — for example an identity-matrix size below 1 or above the supported maximum.	Re-enter the command with an argument in range.
Stack ERROR	The calculation nested too deeply — most often a user-defined f(x)/g(x) that refers back to itself, so the evaluation never bottoms out.	Simplify the expression or remove the circular reference between f and g.
Variable ERROR	SOLVE was started on an expression that contains no variable to solve for (every term is a constant).	Include the unknown in the equation before solving.
Time Out	A ∑, ∏, ∫ or d/dx calculation could not finish within its working range — for example a summation whose index spans far too many terms.	Narrow the range or simplify the summand/integrand.

Can't Solve	SOLVE / the Equation Solver could not find any root from the given starting value.	Store a starting value in the variable closer to the expected root and solve again.
Could not converge	SOLVE / the Equation Solver reached its iteration limit without settling on a root. The best value so far is shown with its residual (L – R).	Press **OK** to **Continue** iterating from that value, or **AC** to **Exit** and keep the near-solution. For a better start, store a value in x closer to the expected root and solve again.

Note

Some apps also show short guidance messages such as **No data**, **No equation**, **f(x) empty** or **Empty**. These are not errors — they simply mean the app needs you to enter its data or expression before it can calculate.

Specifications

Emulator & platforms

Product	eBidyaloy SC-991BF — scientific calculator emulator
Platforms	Windows, macOS, Web, Android and iOS
Calculator apps	13 (Calculate, Statistics, Distribution, Table, Equation, Inequality, Complex, Matrix, Vector, Spreadsheet, Ratio, Math Box, Base-N)
Display	Natural textbook display (MathI/MathO) with an optional enlarged “extended display” for classrooms
Variable memories	A, B, C, D, E, F, x, y, z, M, plus Ans and MatAns
Matrix memories	MatA, MatB, MatC, MatD (up to 4×4)
Physical constants	47 constants in 6 categories (CODATA-based)
Unit conversions	40 conversions in 9 categories
Internal precision	IEEE-754 double precision (~15 significant digits)
Power	None required — the emulator is software and is always ready to use on the host device

Note

Because the SC-991BF is a software emulator, there is no battery to replace and no low-battery warning: it is available whenever the app or web page is open, and its speed depends on the host device.

Frequently Asked Questions

Frequently Asked Questions

Q How can I switch a result between decimal and exact (fraction) form?

A By default (**MathI/MathO**) a result appears in its exact “Standard” form (fraction, π, √). Press ⇄ (FORMAT / S⇔D) to toggle to its decimal value and back. To make results appear as a decimal from the start instead, change **Input/Output** in SETTINGS to **MathI/DecimalO**.

Q What is the difference between Ans memory and variable memory?

A Both store a single value. **Ans** holds the result of the last calculation, letting you carry it straight into the next one (press Ans). A **variable** (A–F, x, y, z, M) is a named container you fill yourself and reuse whenever you need the same value more than once.

Q How can I find a function from an older calculator model?

A Almost every function lives on the **CATALOG**. Press CATALOG to open the catalog menu, then pick the function or constant you need. See **Using the CATALOG** earlier in this guide for the full list.

Q How do I change the calculation result display format?

A Press ⇄ (FORMAT) after a result to toggle exact ⇄ decimal, or open the full FORMAT menu (SHIFT ⇄) for Prime Factor, Recurring Decimal, Sexagesimal, Standard Form and Improper/Mixed Fraction. Result defaults are set in SETTINGS (Number Format, Fraction Result, Complex Result).

Q How can I tell which calculator app I am currently using?

A Press ⌂ (HOME). The icon of the app you are in is highlighted on the HOME screen.

Q How do I calculate $\sin^2 x$?

A $\sin^2 x$ means $(\sin x)^2$. Enter it as the square of the sine — for example $\sin^2 30° = (\sin 30°)^2 = ¼$:
(› sin › 3 › 0 ›) › □² › EXE

Q Why can’t I input i or calculate with complex numbers in Calculate?

A The **Calculate** app works with real numbers. To enter the imaginary unit i or perform complex arithmetic, switch to the **Complex** app.

Q How can I return the calculator to its initial default settings?

A Open any app from HOME, then press ⚙ (SETTINGS) and choose **Reset** ▸ **Settings & Data** ▸ **Yes**. This restores the default settings and clears memories.

Q Do I need to worry about the battery?

A No. The SC-991BF is a software emulator, so it has no battery, never shows a low-battery warning, and is ready whenever you open it.