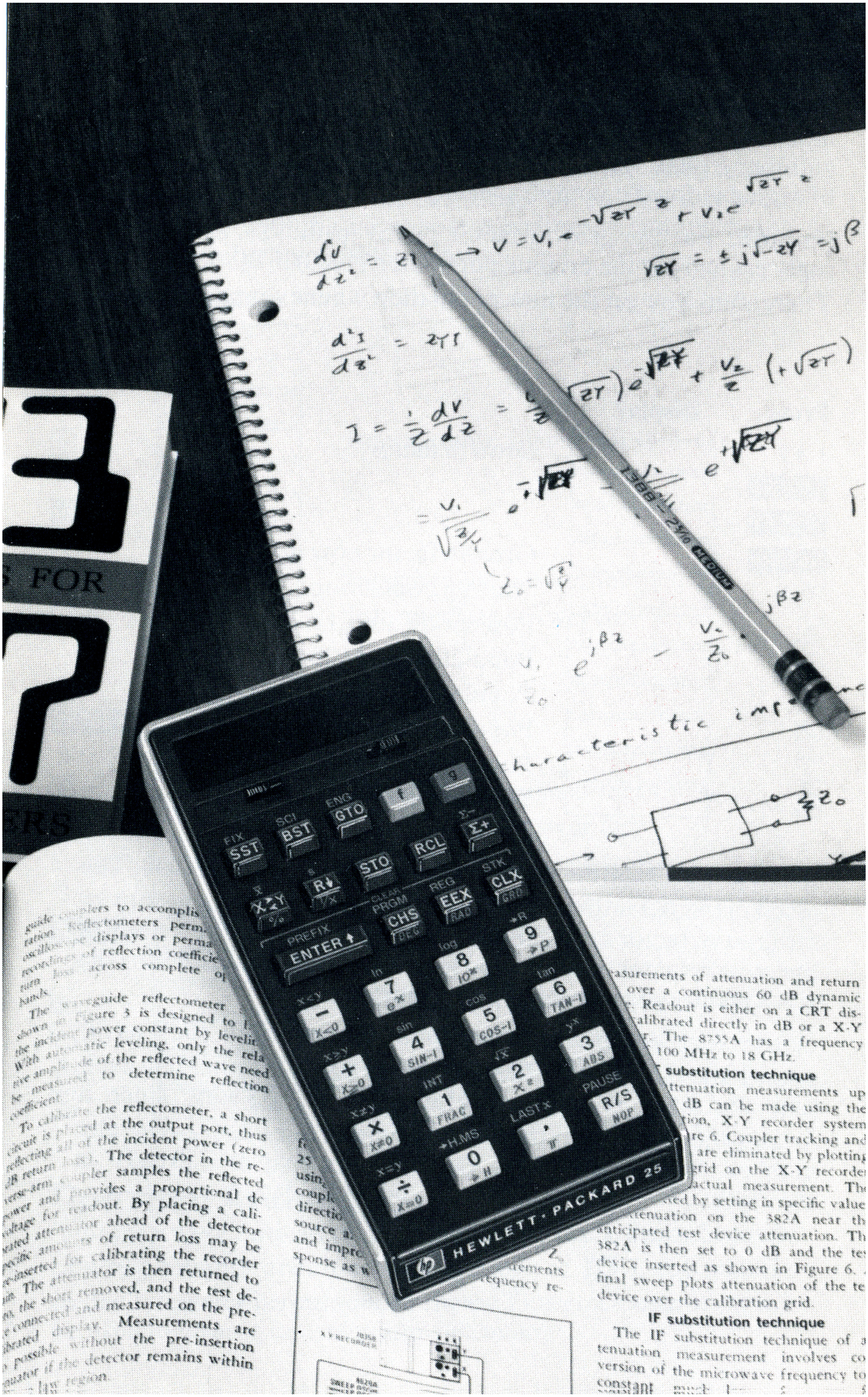

HEWLETT · PACKARD 25

Function Keys

The HP-25 function keys can be used manually or keyed in as part of a program. In this section, each key is individually explained. To use function keys manually, ensure that the PRGM-RUN switch [PRGM ▮▮ RUN] is set to RUN.

LAST X

In addition to the four stack registers that automatically store intermediate results, the HP-25 also contains a separate automatic register, the LAST X register. This register preserves the value that was in the displayed X-register before the performance of a function. To place the contents of the LAST X register into the display again, press [f] [LAST X].

Recovering from Mistakes

[LAST X] makes it easy to recover from keystroke mistakes, such as pressing the wrong function key or keying in the wrong number.

Example: Divide 12 by 2.157 after you have mistakenly divided by 3.157.

Press	Display	
12	12.	
[ENTER↑]	12.00	
3.157 [÷]	3.80	Oops! You made a mistake.
[f] [LAST X]	3.16	Retrieves that last entry.
[×]	12.00	You're back at the beginning.
2.157 [÷]	5.56	The correct answer.

In the above example, when you pressed f LAST X, the contents of the stack and LAST X registers were changed . . .

. . . **from this** . . .

T	0.00		
Z	0.00		
Y	0.00		LAST X
X	3.80		3.16

. . . **to this.**

T	0.00		
Z	0.00		
Y	3.80		LAST X
X	3.16	←	3.16

This made possible the correction illustrated in the example above.

Recovering a Number

The LAST X register is useful in calculations where a number occurs more than once. By recovering a number using LAST X, you do not have to key that number into the calculator again.

Example: Calculate

$$\frac{7.32 + 3.650112331}{3.650112331}$$

Press	Display	
7.32	7.32	
ENTER↑	7.32	
3.650112331	3.650112331	
+	10.97	Intermediate answer.
f LAST X	3.65	Recalls 3.650112331 to X-register.
÷	3.01	The answer.

Prefix Clear

The PREFIX *(clear prefix)* key will clear a blue g prefix key, a gold f prefix key, STO, RCL, or GTO (GTO is explained in section 5, Programming). To clear a prefix you have mistakenly pressed, merely press f PREFIX as the next keystrokes, then press the correct key. For example, to change a blue prefix keystroke to that of another key during a calculation:

Press	Display	
2 g	2.	Oops! You meant to change the sign of the number in the display, but you pressed the blue prefix key by mistake.

f PREFIX	2.00	Clears the blue prefix keystroke.
CHS	-2.00	The correct operation, change sign, is performed.

Number Alteration Keys

Besides CHS there are three keys provided on the HP-25 for altering numbers. These keys are ABS, FRAC and INT, and they are most useful when performing operations as part of a program.

Absolute Value

Some calculations require the absolute value, or magnitude, of a number. To obtain the absolute value of the number in the display, press the g prefix key followed by the ABS *(absolute value)* key. For example, to calculate the absolute value of −3:

Press	Display	
3 CHS	-3.	
g ABS	3.00	|−3|

To see the absolute value of +3:

Press	Display	
g ABS	3.00	|+3|

Integer Portion of a Number

To extract and display the integer portion of a number, press the f prefix key followed by the INT *(integer)* key. For example, to display only the integers of the number 123.456:

Press	Display	
123.456	123.456	
f INT	123.00	Only the integer portion of the number remains.

When f INT is pressed, the fractional portion of the number is lost. The entire number, of course, is preserved in the LAST X register.

Fractional Portion of a Number

To place only the fractional portion of a number into the displayed X-register, press the g prefix key followed by the

FRAC *(fraction)* key. For example, to see the fractional portion of 123.456 used above:

Press	Display	
123.456	123.456	
g FRAC	0.46	Only the fractional portion of the number is displayed, rounded here to normal FIX 2 display.

When g FRAC is pressed, the integer portion of the number is lost. The entire number, of course, is preserved in the LAST X register.

Reciprocals

To calculate the reciprocal of a number in the displayed X-register, then press g 1/x. For example, to calculate the reciprocal of 25:

Press	Display
25 g 1/x	0.04

You can also calculate the reciprocal of a value in a previous calculation without reentering the number. For example, to calculate

$$\frac{1}{1/3 + 1/6}$$

Press	Display	
3 g 1/x	0.33	Reciprocal of 3.
6 g 1/x	0.17	Reciprocal of 6.
+	0.50	Sum of reciprocals.
g 1/x	2.00	Reciprocal of sum.

Square Roots

To calculate the square root of a number in the displayed X-register, press f √x. For example, to find the square root of 16:

Press	Display
16 f √x	4.00

To find the square root of the result:

Press	Display
f √x	2.00

Squaring

To square a number in the displayed X-register, press g x^2 . For example, to find the square of 45:

Press	Display
45 g x^2	2025.00

To find the square of the result:

Press	Display
g x^2	4100625.00

Using Pi

The value π accurate to 10 places (3.141592654) is provided as a fixed constant in the HP-25. Merely press g π whenever you need it in a calculation. For example, to calculate 3π:

Press	Display
3 g π ×	9.42

Example: Trencherman Buck Mulligan looks into a recent edition of the *Guinness Book of Records* and finds that the largest pizza ever baked had a diameter of 21 feet. If his appetite were equal to the task, how many square feet of pizza would Mulligan have to devour in order to consume all of the world's largest pizza?

$$\text{Area} = \pi \left(\frac{d}{2}\right)^2 = \pi \left(\frac{21}{2}\right)^2$$

Press	Display	
21 ENTER↑	21.00	
2 ÷	10.50	
g x^2	110.25	
g π	3.14	
×	346.36	Square feet of pizza.

Pressing g π causes the results in the automatic memory stack to lift.

Percentages

The [%] key is a two-number function that allows you to compute percentages. To find the percentage of a number;

1. Key in the base number.
2. Press [ENTER↑].
3. Key in the number representing percent rate.
4. Press [g] [%].

For example, to calculate a sales tax of 6.5% on a purchase of $1500:

Press	Display	
1500 [ENTER↑]	1500.00	Base number.
6.5	6.5	Percent rate.
[g] [%]	97.50	The answer.

6.5% of $1500 is $97.50.

In the above example, when the [g] [%] keys are pressed, the calculated answer writes over the percentage rate in the X-register, and the base number is preserved in the Y-register.

When you pressed [g] [%], the stack contents were changed . . .

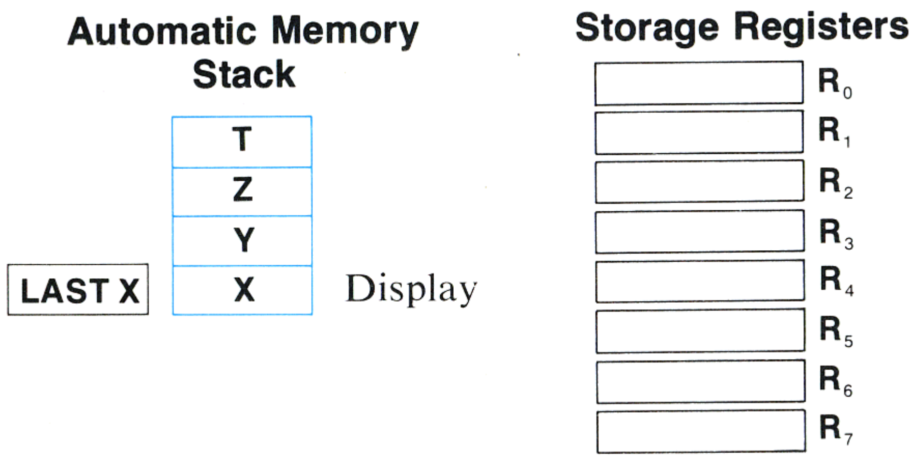
	. . . from this . . .	. . . to this.
T	0.00	0.00
Z	0.00	0.00
Y	1500.00	1500.00
X	6.5	97.50

Since the purchase price is now in the Y-register and the amount of tax is in the X-register, the total amount can be obtained by simply adding:

Press	Display	
[+]	1597.50	Total of price and sales tax combined.

Storage Registers

In addition to automatic storage of intermediate results that is provided by the four-register automatic memory stack, the HP-25 also has eight addressable storage registers that are unaffected by operations within the stack. These storage registers allow you to set aside numbers as constants or for use in later calculations, and they can be used either manually or as part of a program.



The addresses of the storage registers are indicated by number keys [0] through [7], as shown above.

Storing and Recalling Data

To store a value appearing in the display, press **STO** *(store)* followed by a number key ([0] through [7]) specifying the register address where the value is to be stored. For example, to store Avogadro's number (approximately 6.02×10^{23}) in register R_2:

Press	Display	
6.02 **EEX** 23	6.02 23	
STO 2	6.0200000 23	The number is now stored in register R_2.

When a number is stored, it is merely *copied* into the storage register, so 6.02×10^{23} also remains in the displayed X-register.

To copy a number from one of the storage registers into the display, press the **RCL** *(recall)* key followed by the number key of the register address.

For example, to recall Avogadro's number:

Press	Display
CLx	0.00
RCL 2	6.0200000 23

Recalling a number causes the stack to lift unless the preceding keystroke was ENTER↑, CLx or Σ+ (more about Σ+ later).

When you recall a number, it is copied from the storage register into the display, and it also remains in the storage register. You can recall a number from a storage register any number of times without altering it—the number will remain in the storage register as a 10-digit number with a two-digit exponent of 10 until you overwrite it by storing another number there, or until you clear the storage registers.

Example: Three tanks have capacities in U.S. units of 2.0, 14.4, and 55.0 gallons, respectively. If one U.S. gallon is approximately equal to 3.785 liters, what is the capacity of each of the tanks?

Method: Place the conversion constant in one of the storage registers and bring it out as required.

Press	Display	
CLx	0.00	
f FIX 3	0.000	Display mode set.
3.785 STO 0	3.785	Constant placed in register R_0.
2 ×	7.570	Capacity of 1st tank in liters.
14.4 RCL 0 ×	54.504	Capacity of 2nd tank in liters.
55 RCL 0 ×	208.175	Capacity of 3rd tank in liters.
f FIX 2	208.18	Display mode reset.

Clearing Storage Registers

To clear the number from a single storage register, simply store the quantity zero in the register by pressing 0 STO followed by the number key (0 through 7) of the register address.

To clear data from *all* manual storage registers at once, without affecting data in other portions of the calculator, press f REG. This places zero in all eight of the storage registers. Of course, turning the calculator OFF also clears all registers.

Storage Register Arithmetic

Arithmetic is performed *upon the contents of the storage register* by pressing STO followed by the arithmetic function key followed in turn by the register address. For example:

Press	Result
STO [+] 1	Number in displayed X-register added to contents of storage register R_1, and sum placed into R_1: ($r_1 + x \rightarrow R_1$).
STO [−] 2	Number in displayed X-register subtracted from contents of storage register R_2, and difference placed into R_2: ($r_2 - x \rightarrow R_2$).
STO [×] 3	Number in displayed X-register multiplied by contents of storage register R_3, and the product placed into R_3: [(r_3) $x \rightarrow R_3$].
STO [÷] 4	Contents of storage register R_4 divided by number in displayed X-register, and quotient placed into R_4: ($r_4 \div x \rightarrow R_4$).

When storage register arithmetic operations are performed, the answer is written into the selected storage register, while the contents of the displayed X-register and the rest of the stack remain unchanged.

Example: During harvest, a farmer trucks tomatoes to the cannery for three days. On Monday and Tuesday he hauls loads of 25 tons, 27 tons, 19 tons, and 23 tons, for which the cannery pays him $55 per ton. On Wednesday the price rises to $57.50 per ton, and he ships loads of 26 tons and 28 tons. If the cannery deducts 2% of the price on Monday and Tuesday because of blight on the tomatoes, and 3% of the price on Wednesday, what is the farmer's total net income?

Method: Keep total amount in a storage register while using the stack to add tonnages and calculate amounts of loss.

Press	Display	
25 ENTER↑ 27 [+]		
19 [+] 23 [+]	94.00	Total of Monday's and Tuesday's tonnage.
55 [×]	5170.00	Gross amount for Monday and Tuesday.

STO 5	5170.00	Gross placed in storage register R_5.
2 g %	103.40	Deductions for Monday and Tuesday.
STO − 5	103.40	Deductions subtracted from total in storage register R_5.
26 ENTER↑ 28 +	54.00	Wednesday's tonnage.
57.50 ×	3105.00	Gross amount for Wednesday.
STO + 5	3105.00	Wednesday's gross amount added to total in storage register R_5.
3 g %	93.15	Deduction for Wednesday.
STO − 5	93.15	Wednesday deduction subtracted from total in storage register R_5.
RCL 5	8078.45	The farmer's total net income from his tomatoes.

(You could also work this problem using the stack alone, but it illustrates how storage register arithmetic works.)

Storage Register Overflow

If the magnitude of a number in any of the eight storage registers exceeds $9.999999999 \times 10^{99}$, the HP-25 display immediately shows OF *(overflow)* to indicate that a storage register has overflowed.

For example, if you use storage register arithmetic to attempt to calculate the product of 1×10^{50} and 7.5×10^{50} in register R_0, the register overflows and the display shows OF. To see the result of storage register overflow:

Press	Display	
EEX 50	1. 50	
STO 0	1.0000000 50	1×10^{50} placed into storage register R_0.
7.5 EEX 50	7.5 50	
STO × 0	OF	When you multiplied using storage register arithmetic, register R_0 overflowed.

To clear a storage register overflow display, merely press CLx.

Trigonometric Functions

Your HP-25 provides you with six trigonometric functions. It also calculates angles in decimal degrees, radians, or grads; and it converts between decimal degrees and degrees, minutes, seconds.

Trigonometric Modes

When the HP-25 is first turned ON, it "wakes up" with angles specified in decimal degrees. To set radians or grads mode, press the [g] shift key followed by either [RAD] *(radians)* or [GRD] *(grads)*. To switch back to the decimal degrees mode again, press the [g] shift key followed by the [DEG] *(degrees)* key.

Note: 360 degrees = 2π radians = 400 grads

Functions

The six trigonometric functions provided by the calculator are:

[f] [sin] (sine)
[g] [sin⁻¹] (arc sine)
[f] [cos] (cosine)
[g] [cos⁻¹] (arc cosine)
[f] [tan] (tangent)
[g] [tan⁻¹] (arc tangent)

Each trigonometric function assumes angles in decimal degrees, radians, or grads. Trigonometric functions are one-number functions, so to use them you key in the number, then press the function keys.

Example 1: Find the cosine of 35°.

Press	Display	
35	35.	Calculator "wakes up" in decimal degrees mode.
[f] [cos]	0.82	The answer.

Example 2: Find the arc sine in grads of .964.

Press	Display	
.964	0.964	
[g] [GRD]	0.96	Grads mode is set.
[g] [sin⁻¹]	82.87	Grads.

Hours, Minutes, Seconds

The [→H.MS] *(to hours, minutes, seconds)* key converts decimal hours to the format of hours, minutes and seconds. To see the digits for *seconds,* you should specify FIX 4 display format. For example, to convert 12.56 hours to hours, minutes, seconds:

Press	Display	
12.56	12.56	Decimal hours.
[f] [FIX] 4	12.5600	Sets display format.
[f] [→H.MS]	12.3336	This is read as 12 hours, 33 minutes, 36 seconds.

Conversely, the [→H] *(to decimal hours)* key is used to change hours, minutes, seconds into decimal hours. For example, to convert 12 hours, 33 minutes, 36 seconds back into decimal hours:

Press	Display	
[g] [→H]	12.5600	Decimal hours.

Hours to *hours, minutes, seconds* conversion is accurate to 10^{-5} decimal hours.

The [→H] and [→H.MS] keys also permit you to change degrees, minutes, seconds to decimal degrees, and vice versa.

For example, to change 137°45′12″ to decimal degrees:

Press	Display	
137.4512	137.4512	
[g] [→H]	137.7533	Decimal degrees.

The conversion is important because trigonometric functions in the HP-25 operate on angles in *decimal degrees,* but not in *degrees, minutes, seconds.* In order to calculate any trigonometric functions of an angle given in degrees, minutes, seconds, you must first convert the angle to decimal degrees.

Example: Lovesick sailor Oscar Odysseus dwells on the island of Tristan da Cunha (37°03′ S, 12°18′W), and his sweetheart, Penelope, lives on the nearest island. Unfortunately for the course of true love, however, Tristan da Cunha is the most isolated inhabited spot in the world. If Penelope lives on the island of St. Helena (15°55′S, 5°43′W), use the following formula to calculate the great circle distance that Odysseus must sail in order to court her.

$$\text{Distance} = \cos^{-1}\left[\sin(LAT_s)\sin(LAT_d) + \cos(LAT_s)\cos(LAT_d)\cos(LNG_d - LNG_s)\right] \times 60.$$

Where LAT_s and LNG_s = latitude and longitude of the source (Tristan da Cunha).

LAT_d and LNG_d = latitude and longitude of the destination.

Solution: Convert all degrees, minutes, seconds entries into decimal degrees as you key them in. The equation for the great circle distance from Tristan da Cunha to the nearest inhabited land is:

$$\text{Distance} = \cos^{-1}\left[\sin(37°03') \sin(15°55') + \cos(37°03')\cos(15°55')\cos(5°43'W - 12°18'W)\right] \times 60$$

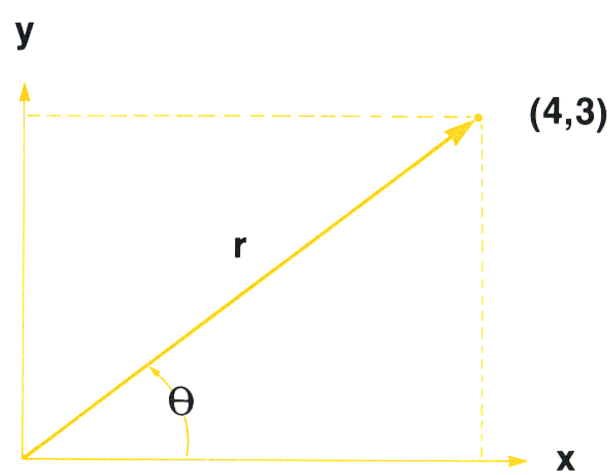
Press	Display	
f FIX 2	0.00	Display mode is set. (Assumes no results remain from previous example.)
g DEG	0.00	Sets decimal degree mode for trigonometric functions.
5.43 g →H	5.72	
12.18 g →H −	–6.58	
f cos	0.99	
15.55 g →H STO 1	15.92	
f cos	0.96	
×	0.96	
37.03 g →H STO 0	37.05	
f cos	0.80	
×	0.76	
RCL 0 f sin	0.60	
RCL 1 f sin	0.27	
×	0.17	
+	0.93	
g cos⁻¹	21.92	
60 ×	1315.41	Distance in nautical miles that Odysseus must sail to visit Penelope.

Polar/Rectangular Coordinate Conversion

Two functions are provided for polar/rectangular coordinate conversion. To convert values in the X- and Y-registers, (representing rectangular x, y coordinates, respectively) to polar r, θ coordinates (magnitude and angle, respectively), press g →P. Magnitude r then appears in the X-register and angle is placed in the Y-register.

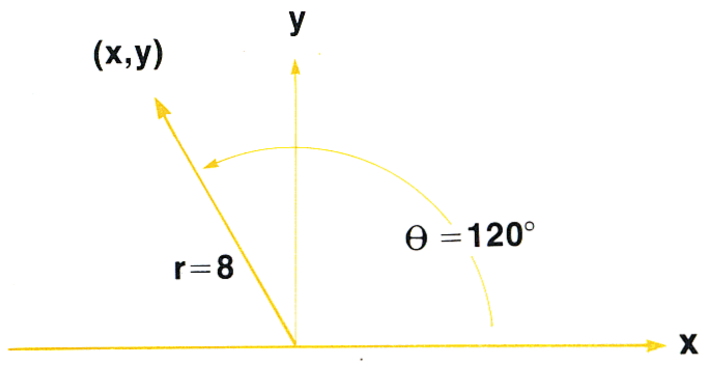
Conversely, to convert values in the X- and Y-registers (representing polar r, θ, respectively) to rectangular coordinates (x, y respectively), press f →R .

Example 1: Convert rectangular coordinates (4,3) to polar form with the angle expressed in radians.



Press	Display	
g RAD	0.00	Specifies radians mode. (Assumes no results remain from previous example.)
3 ENTER↑ 4	4.	Rectangular coordinates placed in X- and Y-registers.
g →P	5.00	Magnitude r.
x⇄y	0.64	Angle θ in radians.

Example 2: Convert polar coordinates (8, 120°) to rectangular coordinates.



Press	Display	
g DEG	0.00	Specifies degrees mode. (Assumes no results remain from previous example.)
120 ENTER↑ 8	8.	Polar coordinates θ and r placed in Y- and X-registers, respectively.
f →R	-4.00	x-coordinate.
x⇄y	6.93	y-coordinate.

Logarithmic and Exponential Functions

Logarithms

The HP-25 computes both natural and common logarithms as well as their inverse functions (antilogarithms):

f ln is $\log_e$ (natural log). It takes the log of the value in the X-register to base e (2.718 . . .) .

g e^x is antilog_e (natural antilog). It raises e (2.718 . . .) to the power of the value in X-register. (To display the value of e, press 1 g e^x.)

f log is $\log_{10}$ (common log). It computes the log of the value in the X-register to base 10.

g 10^x is antilog_{10} (common antilog). It raises 10 to the power of the value in the X-register.

Example 1: The 1906 San Francisco earthquake, with a magnitude of 8.25 on the Richter Scale is estimated to be 105 times greater than the Nicaragua quake of 1972. What would be the magnitude of the latter on the Richter Scale? The equation is

$$R_1 = R_2 - \log \frac{M_2}{M_1} = 8.25 - (\log \frac{105}{1})$$

Solution:

Press	Display	
8.25 ENTER↑	8.25	
105 f log	2.02	
−	6.23	Rating on Richter scale.

Example 2: Ace explorer Jason Quarmorte is using an ordinary barometer as an altimeter. After measuring the sea level pressure (30 inches of mercury) he climbs until the barometer indicates 9.4 inches of mercury. Although the exact relationship of pressure and altitude is a function of many factors, Quarmorte knows that an *approximation* is given by the formula:

$$\text{Altitude (feet)} = 25{,}000\, ln \frac{30}{\text{Pressure}} = 25{,}000\, ln \frac{30}{9.4}$$

Where is Jason Quarmorte?

Solution:

Press	Display	
30 ENTER↑	30.00	
9.4 ÷	3.19	
f ln	1.16	
25000	25000.	
×	29012.19	Altitude in feet.

Quarmorte is probably near the summit of Mount Everest (29,028 ft).

Raising Numbers to Powers

f y^x permits you to raise a positive number (either an integer or a decimal) to any power. For example, calculate 2^9 (i.e., 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2).

Press	Display
2 ENTER↑ 9 f y^x	512.00

Now find $8^{-1.2567}$.

Press	Display
8 ENTER↑	8.00
1.2567 CHS f y^x	0.07

In conjunction with $1/x$, f y^x provides a simple way to extract roots. For example, find the cube root of 5 (This is equivalent to $5^{1/3}$).

Press	Display	
5 ENTER↑	5.00	
3 g 1/x	0.33	Reciprocal of 3.
f y^x	1.71	Cube root of 5.

Example: An aircraft pilot reads a pressure altitude (PALT) of 25,500 feet with a calibrated airspeed (CAS) of 350 knots. What is the flight mach number

$$M = \frac{\text{speed of aircraft}}{\text{speed of sound}}$$

if the following formula is applicable?

M=

$$\sqrt{5\left[\left(\left\{\left[\left(1+0.2\left[\frac{350}{661.5}\right]^2\right)^{3.5}-1\right]\left[1-\left(6.875\times 10^{-6}\right)25{,}500\right]^{-5.2656}\right\}+1\right)^{0.286}-1\right]}$$

Method: The most efficient place to begin work on this problem is at the innermost set of brackets. So begin by solving for the quantity $\left[\frac{350}{661.5}\right]^2$ and proceed outward from there.

Press	Display	
350 ENTER↑ 661.5 ÷	0.53	
g x^2	0.28	Square of bracketed quantity.
.2 × 1 +	1.06	
3.5 f y^x 1 −	0.21	Contents of left-hand set of brackets are in the stack.
1 ENTER↑ 6.875 EEX	6.875 00	
CHS 6 ENTER↑	6.8750000 −06	
25500 × −	0.82	
5.2656 CHS f y^x	2.76	Contents of right-hand set of brackets are in the stack.
× 1 +	1.58	
.286 f y^x 1 −	0.14	
5 × f $\sqrt{x}$	0.84	Mach number of the flight.

In working through complex equations, like the one containing six levels of parentheses above, you really appreciate the value of the Hewlett-Packard logic system. Because you calculate one step at a time, you don't get "lost" within the problem. You see every intermediate result, and you emerge from the calculation confident of your final answer.

Statistical Functions

Summations

Pressing the Σ+ key automatically gives you several different sums and products of the values in the X- and Y- registers at once. In order to make these values accessible for sophisticated statistics problems, they are automatically placed by the calculator into storage registers R_3 through R_7. *The only time that information is automatically accumulated in the storage registers is when the* Σ+ *key is used.* Before you begin any calculations using the Σ+ key, you should first clear the storage registers of data by pressing f REG.

When you key a number into the display and press the Σ+ key, each of the following operations is performed:

- The number that you keyed into the X-register is added to the contents of storage register R_7.
- The square of the number that you keyed into the X-register is added to the contents of storage register R_6.
- The number that you keyed into the X-register is multiplied by the contents of the Y-register, and the product added to storage register R_5.
- The number in the Y-register of the stack is added to the contents of storage register R_4.
- The number 1 is added to storage register R_3, and the total number in R_3 is then written into the display (The stack does not lift).

Thus, each press of the Σ+ key updates these summations and multiplications. The contents of the displayed X-register and the applicable storage registers are as follows:

Register	**Data**	
Displayed X	n	Number of entries.
R_3	n	Number of entries.

R_4	Σy	Summation of y values.
R_5	Σxy	Summation of products of x and y values.
R_6	Σx^2	Summation of x^2 values.
R_7	Σx	Summation of x values.

In addition, the y-value present before the last press of the Σ+ key is retained in the Y-register, while the x-value present before Σ+ was pressed is retained in the LAST X register.

To see any of the summations at any time, you have only to recall the contents of the desired storage register. (In the case of the Σ+ key, recalling storage register contents or keying in a number simply writes over the number of entries (n) that is displayed. The stack does not lift.)

Example: Find Σx, Σx^2, Σy, and Σxy for the paired values of x and y listed below.

y	7	5	9
x	5	3	8

Press	Display	
f REG	0.00	Ensures that all storage registers are cleared to zero initially. (Assumes no results remain from previous example.)
7 ENTER↑	7.00	
5 Σ+	1.00	First pair is summed; $n = 1$.
5 ENTER↑	5.00	
3 Σ+	2.00	Second pair is summed; $n = 2$.
9 ENTER↑	9.00	
8 Σ+	3.00	All the data is summed; $n = 3$.
RCL 7	16.00	Sum of x values from register R_7.
RCL 6	98.00	Sum of squares of x values from register R_6.
RCL 5	122.00	Sum of products of x and y values from register R_5.
RCL 4	21.00	Sum of y values from register R_4.
RCL 3	3.00	Number of entries ($n = 3$).

Mean

The mean (arithmetic average) of data entered and summed using the Σ+ key is available by using the $\bar{x}$ (*mean*) key. When you press f x̄, the mean of the values of x is calculated using the data in storage registers R_3 (*n*) and R_7(Σx) and the formula:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

The easiest way to accumulate the required data in the applicable registers is through the use of the Σ+ key as described above. However, the required data may also be stored directly in storage registers R_3 (*n*) and R_7 (Σx), if desired.

Example: A survey found ten of the wealthiest persons in the United States to have the following ages:

62 84 47 58 68 60 62 59 71 73

To find the average (mean) age of this sample of wealthy persons:

Press	Display	
CLX f REG	0.00	Storage registers and X-register cleared to zero.
62 Σ+, 84 Σ+, 47 Σ+, 58 Σ+, 68 Σ+, 60 Σ+, 62 Σ+, 59 Σ+, 71 Σ+, 73 Σ+	10.00	Number of entries.
f x̄	64.40	Average (mean) age in years.

Standard Deviation

The standard deviation (a measure of dispersion around the mean) is calculated using data in the applicable storage registers and the s (*standard deviation*) key. Pressing f s uses the data in registers R_3 (*n*), R_6 (Σx^2), and R_7 (Σx) to calculate the standard deviation according to the formula:

$$s_x = \sqrt{\frac{\sum x^2 - \frac{(\Sigma x)^2}{n}}{n-1}}$$

For example, to obtain the sample deviation in the above problem:

Press	Display	
f s	10.10	Standard deviation.

If the 10 persons used in the sample were actually *the 10 wealthiest* persons, the data would have to be considered as a population rather than as a sample. The relationship between sample standard deviation (*s*) and the population standard deviation (*s′*) is illustrated by the following equation:

$$s' = s\sqrt{\frac{n-1}{n}}$$

Since *n* is automatically accumulated in register R_3 when the data are accumulated by the Σ+ key, it is a simple matter to convert the sample standard deviation, which has already been calculated, to population standard deviation.

For example, if the accumulations in registers R_3 through R_7 are still intact from the previous example, you can calculate the population standard deviation this way:

Press	Display	
f s	10.10	Sample standard deviation (*s*).
RCL 3	10.00	Recalls *n*.
1 −	9.00	Calculates *n-1*.
RCL 3 ÷	0.90	Divides *n-1* by *n*.
f √x ×	9.58	Population standard deviation (*s′*).

Deleting and Correcting Data

If you key in an incorrect value and have not pressed Σ+, press CLx and key in the correct value.

If one of the values is changed, or if you discover after you have pressed the Σ+ key that one of the values is in error, you can correct the summations by using the Σ− key as follows:

1. Key the *incorrect* data pair into the X- and Y- registers.
2. Press f Σ− to delete the incorrect data.
3. Key in the correct values for x and y. (If one value of an x, y data pair is incorrect, both values must be deleted and reentered.)
4. Press Σ+.

The correct values for mean and standard deviation are now obtainable by pressing f x̄ and f s.

For example, suppose the 62-year old member of the *sample* as given above were to lose his position as one of the wealthiest persons because of a series of ill-advised investments in cocoa futures. To account for the change in data if he were replaced in the sample by a 21-year old rock musician:

Press	Display	
62	62.	Data to be replaced.
f Σ−	9.00	Number of entries (n) is now nine.
21	21.	The new data.
Σ+	10.00	Number of entries (n) is ten again.

The new data has been calculated into each of the summations present in the storage registers. To see the new mean and standard deviation:

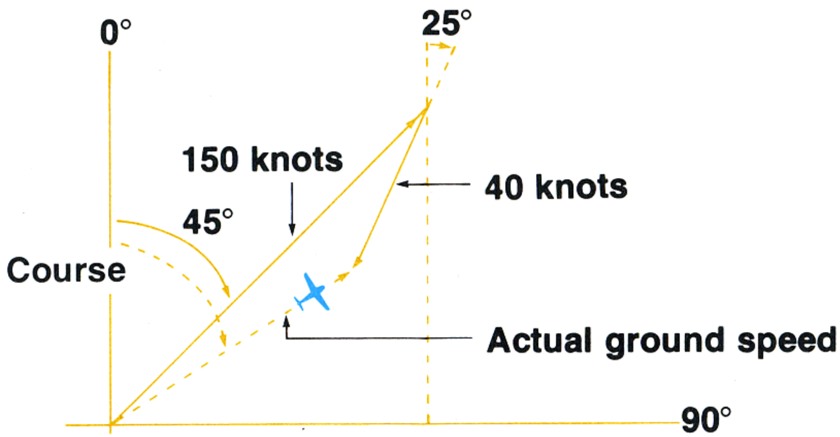
Press	Display	
f x̄	60.30	The new average (mean) age.
f s	17.09	The new standard deviation.

Vector Summations

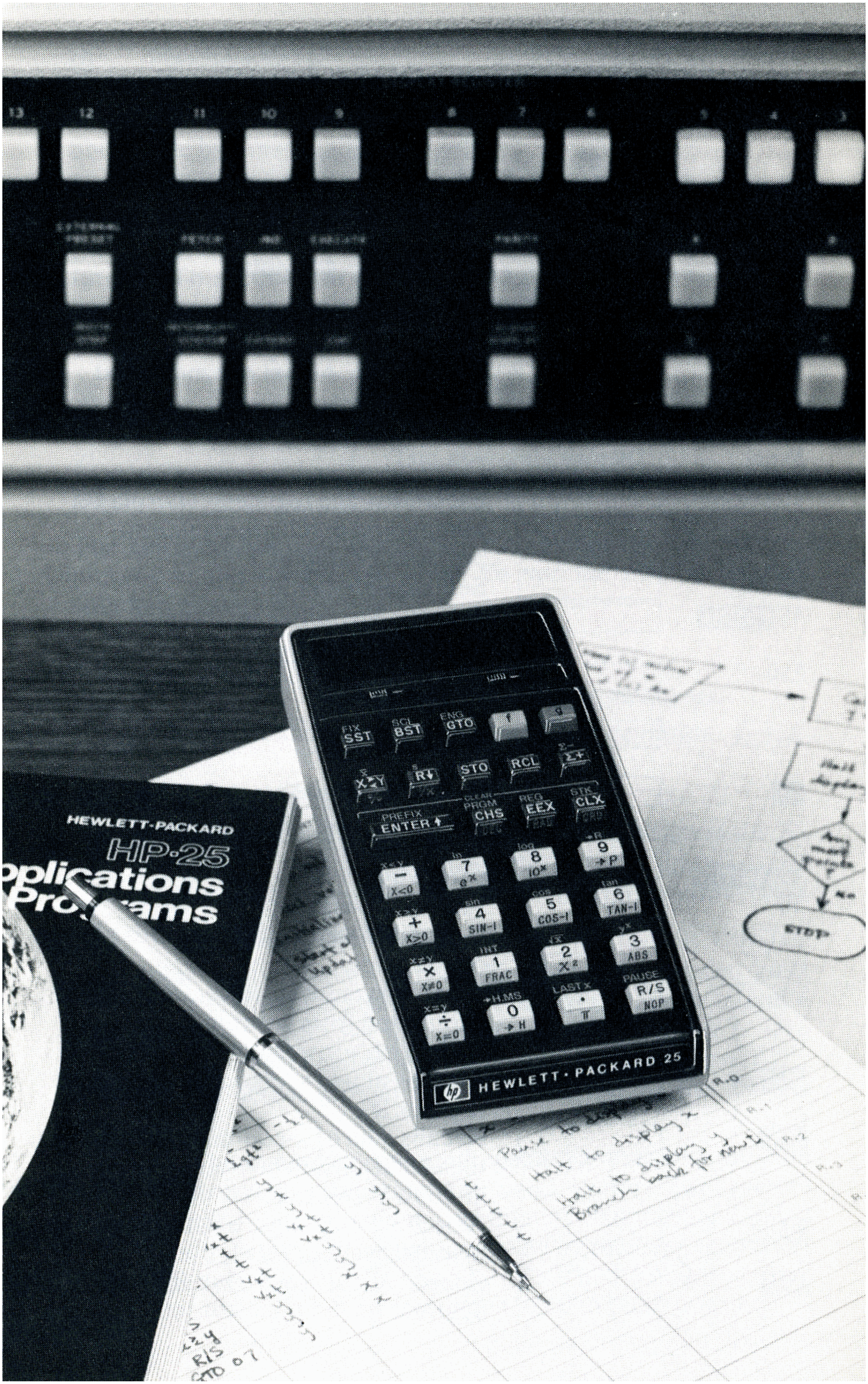
The Σ+ key can be used to sum any quantities that are in the X- and Y-registers. You can even perform vector addition and subtraction using rectangular to polar coordinate conversion and the Σ+ and Σ− keys.

Example: In his converted Swordfish aircraft, grizzled bush pilot Apeneck Sweeney reads an air speed of 150 knots and a heading of 045° from his instruments. The Swordfish is also being buffeted by a headwind of 40 knots from a bearing of 025°. What is the actual ground speed and course of the Swordfish?

Method: The course and ground speed are equal to the sum of the instrument vector and the wind vector. The vectors are converted to rectangular coordinates and summed using the Σ+ and Σ− keys. Their sum is recalled by recalling the values in storage registers R_4 ($\sum y$) and R_7 ($\sum x$), and the new rectangular coordinates are then converted back to polar coordinates to give the vector of the actual ground speed and course.



Press	Display	
f REG	0.00	Clears storage registers. (Display assumes no results remain from previous examples.)
g DEG	0.00	Sets degrees mode.
45 ENTER↑	45.00	θ for the Swordfish instrument vector.
150	150.	r for the Swordfish instrument vector.
f →R	106.07	Converted to rectangular coordinates.
Σ+	1.00	Instrument coordinates accumulated in storage registers R_4 and R_7.
25 ENTER↑	25.00	θ for wind vector.
40	40.	r for wind vector.
f →R	36.25	Converted to rectangular coordinates.
f Σ−	0.00	Coordinates for wind vector subtracted from coordinates for Swordfish's instrument vector.
RCL 4	89.16	Recalls sum of y-coordinates from register R_4.
RCL 7	69.81	Recalls sum of x-coordinates from register R_7. (Sum of y-coordinates lifted to Y-register.)
g →P	113.24	Actual ground speed of the Swordfish in knots.
x⇄y	51.94	Course of the Swordfish in degrees.


HEWLETT-PACKARD
HP-25
HEWLETT • PACKARD 25

Section 5

Programming

As we briefly explained in the introduction, calculator programming is as simple as pressing the keys you would manually press to solve your problem. But even though HP-25 calculator programming is simple to understand and use, it is very powerful, featuring:

- An obvious programming language.
- 49 usable steps of program memory.
- The ability to combine several keystrokes into each step.
- Decision-making capability for sophisticated routines.
- Several editing operations to facilitate corrections.

Together these features provide you with the tools necessary to tackle complex problems with unabashed confidence.

What is a Program?

A program is nothing more than a sequence of manual keystrokes that is remembered by the calculator. You can then execute the program as often as you like with less chance of error. The answer displayed at the end of execution is the same one you would have obtained by pressing the keys one at a time manually. No prior programming experience is necessary for HP-25 calculator programming.

Why Write Programs?

Programs are written to save you time on repetitive calculations. Once you have written the keystroke procedure for solving a particular problem and recorded it in the calculator, you need no longer devote attention to the individual keystrokes that make up the procedure. You can let the calculator solve each problem for you. And because you can easily check the procedure in your program, you have more confidence in your final answer since you don't have to worry each time about whether or not you have pressed an incorrect key. The calculator performs the drudgery, leaving your mind free for more creative work.

Three Modes of Operation

There are three ways to use your HP-25 calculator:

1. Manual RUN mode PRGM RUN
2. PRGM mode PRGM RUN
3. Automatic RUN mode PRGM RUN

Manual RUN Mode

The functions and operations you have learned about in the first four sections of this handbook are performed manually one at a time with the PRGM-RUN switch set to RUN PRGM RUN. These functions combined with the automatic memory stack enable you to calculate any problem with ease.

PRGM Mode

In PRGM *(program)* mode the functions and operations you have learned about are not executed, but instead are recorded in a part of the calculator called *program memory* for later execution. All operations on the keyboard except three can be recorded for later execution with the PRGM-RUN switch set to PRGM PRGM RUN. The three operations that cannot be recorded are:

SST

BST

f PRGM

These three operations work in PRGM mode to help you write and record your programs.

Automatic RUN Mode

The HP-25 can also be used to automatically execute a list of operations with the PRGM-RUN switch set to RUN PRGM RUN if they have previously been recorded in program memory. Instead of your having to press each key manually, the recorded operations are executed sequentially in automatic RUN mode when you press R/S *(run/stop)*. You press only one key and the entire list of recorded operations is executed much more quickly than you could have executed them yourself.

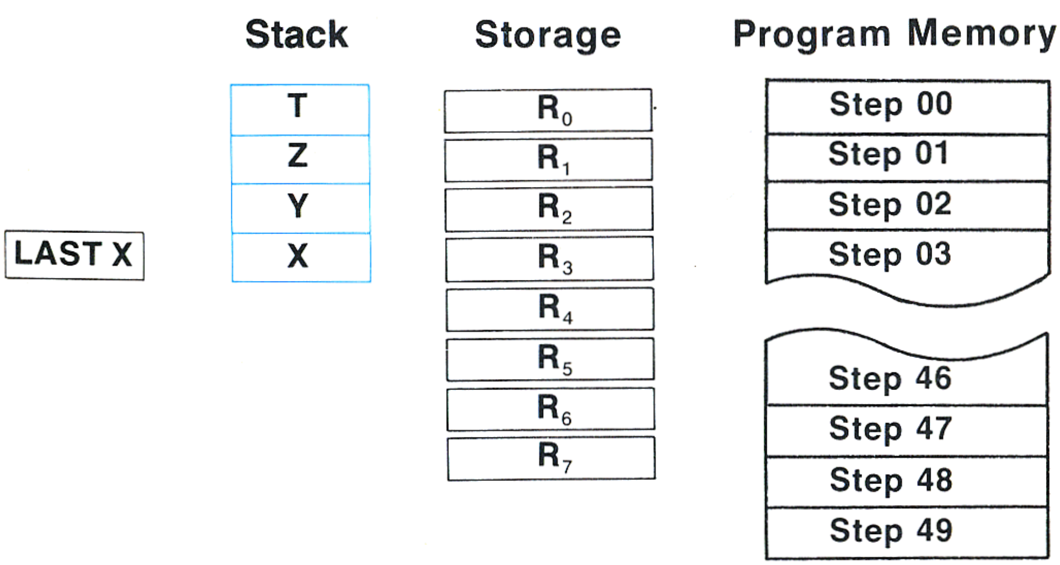
Introductory Program

The area of a sphere program you wrote, recorded, and executed in the introduction showed you that the sequence of keystrokes used to solve a problem manually is the same sequence used in a program. Now let's return our attention to that program to explain the information displayed in PRGM mode.

First, set the PRGM-RUN switch to PRGM PRGM RUN so that the sequence of keystrokes will be recorded for later execution. Second, press f PRGM to clear the calculator of previous programs. The display will show:

00

This tells you that you are at the beginning of program memory. Step 00 contains an automatic stop instruction and cannot be used to record your program keystrokes. Program keystrokes are recorded in steps 01 through 49. (See figure below.)



As you can see, the program memory for the HP-25 is separate from the four stack registers, the LAST X register, and the eight storage registers.

With [00] displayed in PRGM mode, you are ready to key in your program. Surface area of a sphere is calculated using the formula $A = \pi d^2$. The short list of keys for the area of a sphere program is shown below:

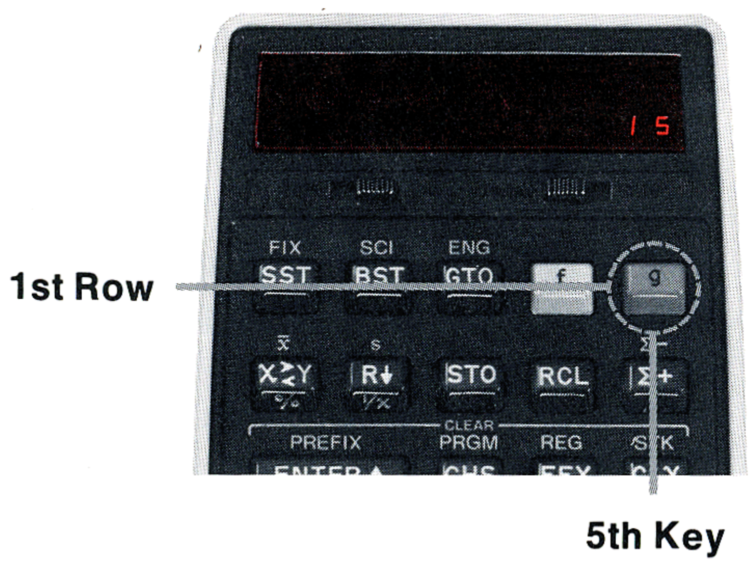
Keys	Comments
g x^2	These keys square the diameter.
g π	These keys place π in the X-register.
×	This key multiplies d^2 by π.

Keycodes

Press the first key of the program and the display will change to:

[15]

The two numbers on the right of the display designate the key stored in that step. Each key on the keyboard has a two-digit keycode. For convenience, the digit keys are coded 00 through 09. All other keys are coded by their position on the keyboard. The first digit denotes the row of the key and the second digit the number of the key in that row. So 15 tells you that the key is in the first row on the calculator and that it is the fifth key in that row, the g key.



This handy matrix system allows you to easily determine the code for each instruction without using a reference table.

Merged Keycodes

To conserve program memory when using prefixed functions, the keycodes for the prefix and the function are merged into one step. For an example of this press the second key of the program, [x^2], and the display will change to:

01 15 02

The two-number code 01 that has appeared on the left side of the display designates the step number of program memory that is being displayed. The two pairs of numbers on the right side of the display indicate that the function [g] [x^2] has been recorded in that step (01) of program memory. Digits 1 and 5 denote the [g] key. Digits 0 and 2 denote the [2] key. The operation stored then, is [g] [2] which is the x^2 function. In every case, a single operation (e.g., [f] [sin], [STO] [+] [1], [x⇄y]) uses only one step of program memory.

> Each operation, prefixed or not, requires only one step of program memory.

The keys for finding the area of a sphere and their corresponding displays are shown below. Press each key in turn and verify the keycode shown in the display.

Key	Display
[g] [x^2]	01 15 02
[g] [π]	02 15 73
[×]	03 61

In this case, a program consisting of five keystrokes takes only three steps of program memory.

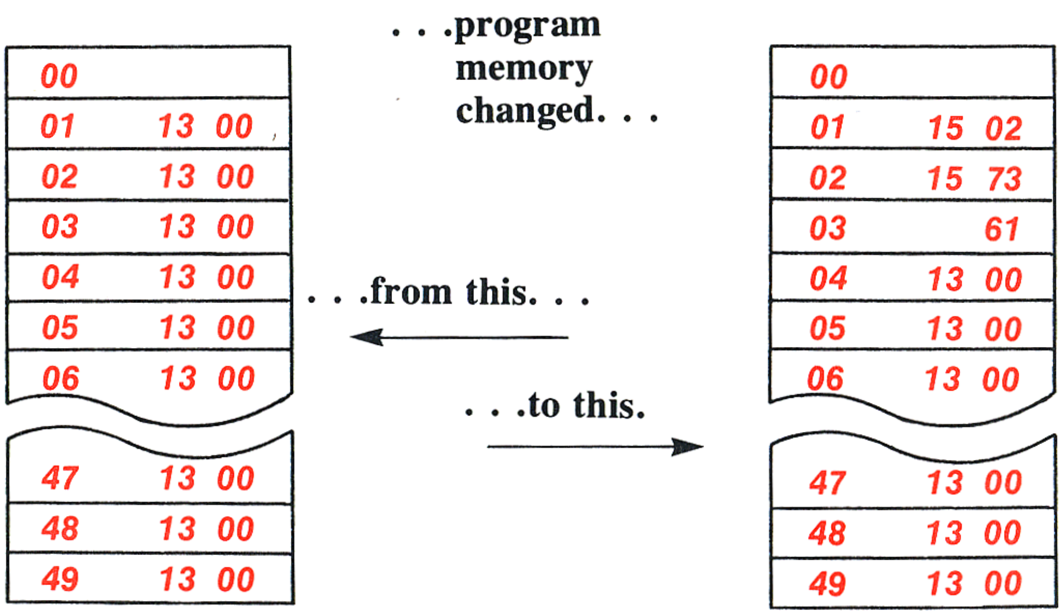
Running a Program

Programs are executed in automatic RUN mode. So first set the PRGM-RUN switch to RUN PRGM [switch] RUN. Next press GTO 0 0. This operation resets the calculator so that program execution will begin from step 00 (Pressing f PRGM in RUN mode accomplishes the same thing.) Then, key in a value for a diameter and press R/S in RUN mode to run your program. The operations stored in program memory are executed sequentially downward from step 00. First step 01 is executed, then step 02, then step 03, and then step 04, which now contains a special instruction, GTO 0 0.

GTO 00

The GTO 0 0 instruction in step 04 is not an instruction you keyed in yourself. It was already there. If you press f PRGM in PRGM mode or if you switch the calculator OFF and ON again, program memory is *filled* with GTO 0 0 instructions. The three-step program you keyed in replaced three of these instructions. Program memory was changed as shown in the following illustration.

When you keyed in your program. . .



The illustration on the left shows program memory immediately after pressing [f] [PRGM] in PRGM mode or turning the HP-25 ON. The illustration on the right shows program memory after recording the three-step example program.

A [GTO] [0] [0] instruction in the program tells the calculator to go to step 00 and execute the automatic stop instruction there next. If [R/S] is pressed again in automatic RUN mode, the calculator will begin executing instructions from step 00 as it did the first time. Each time the calculator executes the program, it ends execution at step 00, ready to begin again.

If you had recorded a 49-step program, after executing step 49 the calculator would execute the automatic stop instruction stored in step 00. Then you would have to press [R/S] to execute the program again.

Now try an example.

Example. Calculate the surface area of a spherical "cat's-eye" (marble) with a diameter of 1.3 centimeters. Then calculate the surface area of a baseball with a diameter of 2.5 inches.

Press	Display	
1.3 [R/S]	5.31	Area of the marble in square centimeters.
2.5 [R/S]	19.63	Area of the baseball in square inches.

Each time you press [R/S] the calculator executes the sequence of keystrokes you have recorded. You calculate the same answers you would obtain if you did each problem manually, but without the time or the tedium.

Writing a Second Program

Now let's write a second program and use it to further explore the programming capability of your HP-25 calculator. Suppose you want to write a program that will calculate the increase in volume of a spherical balloon as its diameter increases using the formula:

$$\text{Increase in volume} = 1/6\,\pi\,(d_1^3 - d_0^3),$$

where d_0 is the original diameter of the balloon and d_1 is the new diameter. If d_0 were entered in the Y-register and d_1 were keyed into the X-register, the problem could be solved manually by pressing the keys shown in the left-hand column that follows.

Keys	Display	
ENTER↑	01 31	
3	02 03	
f yˣ	03 14 03	Cube the new diameter.
x⇄y	04 21	
3	05 03	
f yˣ	06 14 03	Cube the original diameter.
−	07 41	Subtract the cubes.
g π	08 15 73	
×	09 61	Multiply by π.
6	10 06	
÷	11 71	Divide by 6.

The program keystrokes for this problem are the same. Simply switch to PRGM mode PRGM ▮ RUN and press f PRGM to clear program memory and display step 00. Then key in the list of keys above. The keys are not executed, but are recorded in program memory steps 01 through 11. Verify that each keycode is correct as you key in each instruction by checking the displays shown.

(Notice that you had to record the ENTER↑ key as an instruction in this program. The ENTER↑ instruction here separates the number 3 that is the second step of the program from the digits for the new diameter that you will key in later.)

To run the program switch to automatic RUN mode PRGM ▮▯ RUN and press f PRGM (or GTO 0 0) so that the calculator will begin execution from step 00. Then try the following example.

Example. Find the increase in volume of a spherical balloon if the diameter changes from 30 feet to 35 feet.

Press	Display	
30 ENTER↑	30.00	Enter the original diameter into Y.
35 R/S	8312.13	Key the new diameter into X and run the program. The answer, in cubic feet, is displayed.

Displaying Each Step

In order to look at this program, you need to be able to display each step. Two operations allow you to do this: SST *(single step)* and BST *(back step)*.

With the increase in sphere volume program still recorded in the calculator set the PRGM-RUN switch to RUN PRGM ▮▯ RUN and press f PRGM to reset the calculator to step 00. Then switch to PRGM mode PRGM ▯▮ RUN and press SST once. The display will change to:

01 31

Press SST again and the display will change to:

02 03

Now press BST. You can see what has happened. You are back at program memory step 01. Press BST again and step 00 is displayed. Pressing BST again does nothing.

SST displays the contents of the *next* step of program memory.

BST displays the contents of the *previous* step of program memory.

Of course, because these two keys work in PRGM mode, neither can be stored in program memory.

Displaying a Particular Step

If you want to see one of the later steps of your program, SST is not convenient. To display a particular step of program memory use the GTO key with the PRGM-RUN switch set to RUN PRGM RUN. Simply press GTO and then key in the desired two-digit step number. Then set the PRGM-RUN switch to PRGM PRGM RUN and the contents of the specified step will be displayed.

For example, to see step 10 in the previous program, set the PRGM-RUN switch to RUN PRGM RUN and press GTO [1] [0]. Then switch back to PRGM mode PRGM RUN. The display will show:

10 06

When using the GTO key in this way, always use two digits for designating step numbers. For instance, to see step 6 you must press GTO [0] [6] in RUN mode and then switch back to PRGM mode.

If the first digit key following GTO is greater than four, the GTO key is ignored and the number is keyed into the X-register. Similarly, if one of the two keys following GTO is not a digit key, the GTO key is ignored and the operation associated with the invalid key is performed.

Interrupting Program Execution

From time to time you will want a program to stop execution by itself so that you can enter new data or view an intermediate result. There are two operations on your HP-25 calculator that will automatically interrupt program execution when they are encountered as program instructions: [R/S] and f [PAUSE].

Stopping Program Execution

[R/S] works differently as an executed instruction in a program than it does when pressed from the keyboard. As an executed instruction, [R/S] stops program execution, allowing you to key in new data or to write down an intermediate result.

When [R/S] is then pressed from the keyboard in automatic RUN mode, the calculator continues execution sequentially downward.

Example Program. Universal Tins, a canning company, needs to calculate the volumes of various cylindrically-shaped cans. Universal would also like to be able to record the area of the base of each can before the volume is calculated. One program to solve this problem follows.

This program calculates the area of the base of each can and then stops. When after you have written down that result, the program can be restarted to calculate the final volume. The formula used is:

$$\text{Volume} = \text{base area} \times \text{height} = \pi r^2 \times h$$

The radius *(r)* and the height *(h)* of the can are keyed into the X- and Y-registers, respectively, before the program is run.

To record this program, set the PRGM-RUN switch to PRGM PRGM [switch] RUN and press [f] [PRGM] to clear program memory and display step 00. Then key in the following list of keys.

Press	Display	
[g] [x^2]	01 15 02	Square the radius.
[g] [π]	02 15 73	Place π in X.
[×]	03 61	Calculate the area of the base.
[R/S]	04 74	Stop to record the area.
[×]	05 61	Calculate the final volume.

In order to run this program set the PRGM-RUN switch to RUN PRGM [switch] RUN and press [f] [PRGM] so that the calculator will begin execution from step 00. Then use the program to complete the table below:

Height	Radius	Area of Base	Volume
25	10	?	?
8	4.5	?	?

Press	Display	
25 ENTER↑	25.00	Enter the height into the Y-register.
10 R/S	314.16	Program stops to display the area.
R/S	7853.98	Volume of first can is calculated.
8	8.00	Enter the height into the Y-register.
ENTER↑		
4.5 R/S	63.62	Program stops to display the area.
R/S	508.94	Second volume is calculated.

With the height in the Y-register and the radius in the X-register, pressing R/S in automatic RUN mode calculates the area of the can's base; the program stops at the first R/S instruction encountered. Pressing R/S again calculates the volume of the can and program execution stops at step 00, ready to run again.

In general, R/S is recorded into a program when you need to display *more* than one answer. To display *only one* answer or the final answer of a series, the GTO 0 0 instruction in a program is more convenient since the calculator ends execution at step 00, ready to begin again.

Pausing During Program Execution

An f PAUSE instruction executed in a program momentarily interrupts program execution to display intermediate results that do not have to be written down. The length of the pause is about one second, although more than one f PAUSE instruction can be used to lengthen the time if desired.

To see how f PAUSE can be used in a program, we'll modify the cylinder volume program in the previous example. In the new program the area of the base will only be briefly displayed before the volume is calculated. This example will also show how different programming approaches can be taken to solve the same problem.

To key in the program, set the PRGM-RUN switch to PRGM PRGM ▮▮ RUN and press f PRGM to clear program memory and display step 00. Then key in the following list of keys.

Press	Display	
g x^2	01 15 02	Squares the radius in X.
g π	02 15 73	Places π in X.
×	03 61	Calculates the area of the base.
f PAUSE	04 14 74	Pauses to show the base area for one second.
×	05 61	Calculates final volume of can.

This program also assumes the height has been entered into the Y-register and the radius has been keyed into the X-register. If you have stored the instructions, set the PRGM-RUN switch to RUN PRGM ▮▮ RUN and press f PRGM so that the calculator will begin execution from step 00. Now complete the table below using the new program.

Height	Radius	Area of Base	Volume
20	15	?	?
10	5	?	?

Press	Display	
20 ENTER↑	20.00	Enter the height into the Y-register.
15 R/S	706.86	Area of base is displayed for one second.
	14137.17	Program stops, displaying the volume.
10 ENTER↑	10.00	Enter the second height into Y.

5 [R/S]	78.54	Area of base is displayed for one second.
	785.40	Program stops, displaying the volume.

Program Stops

At times a mistake of some kind in your program will stop program execution. To help you identify why the calculator stopped in the middle of your program, possible reasons are listed below.

Executing a R/S. The execution of a [R/S] instruction in a program halts program execution at the step following the [R/S].

Executing Step 00. Whenever step 00 is executed in a program, program execution stops at step 00.

Pressing Any Key. Pressing any key halts program execution. Be careful to avoid pressing keys during program execution. If a program has been stopped by pressing a key, be careful not to restart program execution in the middle of a digit entry key sequence within the program. For example in the section of a program shown below, if program execution halted at step 23, the number 13 would appear in the display. If [R/S] is pressed, the number 13 would be automatically pushed up into the stack and the number 4.7 would be keyed into the X-register.

19	61	
20	14 03	
21	01	Digit entry
22	03	Digit entry
23	04	Digit entry
24	73	Digit entry
25	07	Digit entry
26	15 22	

To avoid problems like this, you should switch to PRGM mode to see whether or not you are in the middle of a digit entry key sequence. If you are, you should use [SST] or [BST] to correct the situation. In this case, you should press [BST] twice in PRGM mode, then switch back to RUN mode and press [CLx]. Finally you can press [R/S] to resume program execution.

Overflow Calculations. Your HP-25 has been designed so that by looking at the display you can always tell why the calculator stops. If program execution stops because the result of a calculation in the X-register is a number with a magnitude greater than $9.999999999 \times 10^{99}$, all 9's are displayed with appropriate sign. It is then easy to determine the operation that caused the overflow by switching to PRGM mode and identifying the keycode in the display.

If the overflow occurs in one of the storage registers, possibly the result of storage register arithmetic or the summations with Σ+ ,the calculator will display OF to inform you of the overflow. Check the storage registers to see in which register the overflow has occurred.

If the result of a calculation is a number with a magnitude less than 10^{-99}, zero will be substituted for the number and a running program will continue to execute normally.

Improper Operation Stops. Calculations that cause the word Error to be displayed also stop program execution. You can identify the reason for the stop by switching momentarily to PRGM mode to see the keycode of the improper operation. A list of improper operations can be found in appendix B.

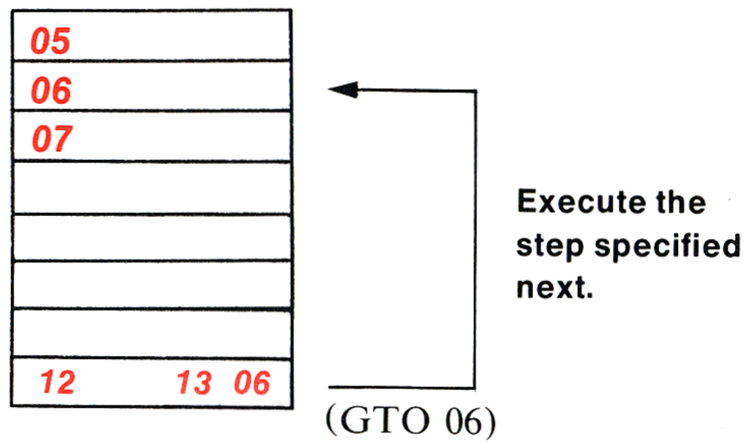
Branching

Although program execution is normally sequential, with one step executed after another, execution can be transferred or "branched" to any step in program memory. The "branch" can be made unconditionally or it can be made dependent on the outcome of a comparison of data values.

Unconditional Branching

You have seen how GTO is used in manual RUN mode to help you display any step in program memory. As an instruction executed in a program GTO is used to branch program execution to the step number specified. It can tell the calculator to execute step 00 next, as we have already seen, or to execute any other step in program memory.

Unconditional Branching



When recording an unconditional branch always follow the GTO key with two digit keys to designate the step number. For instance, to branch to step 6 the program instruction must be GTO [0] [6].

If the first digit key following GTO is greater than four, the GTO key is ignored and the number is stored in that step of program memory. Similarly, if one of the two keys following GTO is not a digit key, the GTO key is ignored and the invalid key is stored in program memory.

Example Program. The following program is an interesting one to show your friends. It calculates the squares of consecutive whole numbers beginning with zero. The calculator continues to compute the square of the next consecutive whole number until you press [R/S] to stop program execution (or until the calculation overflows). The simple formula used is: $x \times n^2$ where n is continually incremented by one.

To key in the program set the mode switch to PRGM (PRGM ■ RUN) and press f PRGM to clear program memory and display step 00. Then key in the list of keys shown below.

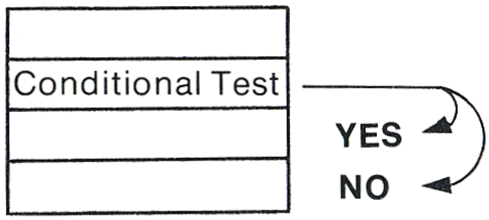
Press	Display	
0	01 00	
STO 1	02 23 01	Store zero in R_1.
RCL 1	03 24 01	Recall the current number for squaring.
g x^2	04 15 02	Square the number.
f PAUSE	05 14 74	Display the square briefly.
1	06 01	
STO + 1	07 23 51 01	Increment the number in R_1 by one.
GTO 0 3	08 13 03	Transfer program execution to calculate the next square.

The program calculates the square of the number in storage register R_1, starting with zero. It pauses to show the answer and then increments the contents of the register by one. The unconditional branch at the end of the program is used to transfer program execution back to step 03 so that the calculation can be repeated with the new value in register R_1.

To run the program set the PRGM-RUN switch to RUN (PRGM ■ RUN) and press f PRGM so that the calculator will begin execution from step 00. Then simply press R/S. The squares of consecutive whole numbers will be shown one by one in the display. Press R/S again to stop execution whenever you wish.

Conditional Branching

Eight different program instructions give the HP-25 the ability to make decisions within a program depending on the outcome of a comparison of data values. These "conditionals" transfer program execution based on the outcome of the test. If the answer is YES, program execution continues sequentially downward. If the answer is NO, the calculator branches around the following step, which can contain an unconditional branch or a simpler instruction (CHS, for example). The program makes a decision for you!



Program execution branches around one step if the answer to the test is NO.

The eight different conditionals in your HP-25 are shown here. In each case, the tests are made on the 10-digit numbers and two-digit exponents actually stored in the stack registers, not on the displayed values.

f x<y tests to see if the value in the X-register is less than the value in the Y-register.

f x≥y tests to see if the value in the X-register is greater than or equal to the value in the Y-register.

f x≠y tests to see if the value in the X-register is not equal to the value in the Y-register.

f x=y tests to see if the value in the X-register is equal to the value in the Y-register.

g x<0 tests to see if the value in the X-register is less than zero.

g x≥0 tests to see if the value in the X-register is greater than or equal to zero.

g x≠0 tests to see if the value in the X-register is not equal to zero.

g x=0 tests to see if the value in the X-register is equal to zero.

Example Program. This program calculates the arc sine of an input value x (x must be within the limits of -1 and $+1$). The program tests the resulting angle, and if it is not greater than zero, adds 360 degrees to it. The angle displayed by the program, then, is always positive.

To key in the program set the mode switch to PRGM PRGM ▯▮ RUN and press f PRGM to clear program memory and display step 00. Then key in the following list of keys.

Press	Display	
g sin⁻¹	01 15 04	Calculates the arc sine.
g x≥0	02 15 51	Compares the result to zero.
GTO 0 0	03 13 00	If greater than zero, display arc sine.
3	04 03	Otherwise
6	05 06	add
0	06 00	360 degrees
+	07 51	to the arc sine.

To run the program set the PRGM-RUN switch back to RUN PRGM RUN and press f PRGM so that the calculator will begin execution from step 00. Then key in positive or negative values for x. The resultant arc sine will always be positive.

Press	Display	
g DEG	0.00	Set degrees mode.
.5 R/S	30.00	Arc sine of .5 equals 30 degrees.
.5 CHS	-0.5	Key in negative value for x.
R/S	330.00	360 is added to the arc sine to give a positive angle.

Editing a Program

Even the most experienced programmer finds errors in his programs. These errors range from mistakes in the original equations to mistakes in recording the program. Wherever they occur they need to be found and corrected, and the HP-25 is designed to make this error-checking process as easy as possible.

Finding the Error

One of the easiest ways to find out if your program is working properly is to work a test case in which you either know the answer or the answer can be easily determined. For example, if you have a program that calculates the area of a circle using the formula $area = \pi \times r^2$, you can easily determine that an input value of 1 for r will give an answer of π.

SST Execution. In longer programs a wrong test-case answer will seldom pinpoint the mistake. For these cases, you can slow down program execution by using the SST key in RUN mode. In RUN mode, the SST key will execute your program instructions one at a time. When you hold the SST key down in RUN mode, the program step number and keycode are displayed. When you release the SST key, the instruction is executed. Use SST on the simple area of a circle program shown below to familiarize yourself with its operation.

Example Program. This program calculates the area of a circle using the formula: $A = \pi r^2$ where r is the radius. Set the PRGM-RUN switch to PRGM (PRGM ▮▮ RUN) and press f PRGM to clear program memory and display step 00. Then key in the list of keys shown below.

Press	Display
g x^2	01 15 02
g π	02 15 73
×	03 61

The program assumes that a value for r has been keyed into the X-register. To run the program, set the PRGM-RUN switch back to RUN (PRGM ▮▮ RUN) and press f PRGM. Now step through the program in slow motion using a value of 10 for r.

Press	Display	
10	10.	
SST	01 15 02	When you hold SST down, the first instruction is displayed.
	100.00	When you release SST, the first instruction is executed.
SST	02 15 73	Again holding SST down displays the second instruction.
	3.14	Again releasing SST executes the second instruction.
SST	03 61	Holding SST down displays the third instruction this time.
	314.16	And releasing SST executes the third instruction.

You can see that it would be easy to spot a mistake in your program using the SST key.

When you hold the BST key down in RUN mode, the program step number and keycode for the previous step are displayed. When you release BST, the X-register is again displayed. However if you switch back to PRGM mode, you will find that the previous step is now displayed. And if you press R/S in RUN mode after pressing BST, the calculator will begin execution from the previous step in program memory. Now press BST in RUN mode to review the program instructions of the above program.

Press	Display	
BST	03 61	Holding BST down in RUN mode displays the previous instruction.
	314.16	Releasing the BST key displays the original contents of the X-register.
BST	02 15 73	Again holding BST down displays the previous step in program memory.
	314.16	And releasing BST displays the original contents of the X-register again.

If you now switch to PRGM mode the second step will be displayed:

02 15 73

Cued Stops. If you have a program that is halted several times during execution for data entries, you may want to "identify" each stop by recording a familiar number into the program just before each R/S instruction. Then when the calculator stops execution because of the R/S instruction in the program, you can look at the displayed X-register to see the "identification number" for the required input. For example if your program contains eight stops for data inputs, it may be helpful to have the numbers 1 through 8 appear so you know which input is required each time. These identification numbers are helpful in editing a program.

If you key in data after the program has stopped running, remember that resuming program execution does not terminate digit entry. Thus, the calculator will assume that the digits in the program are part of the number you have just keyed in unless you press ENTER↑ after you key in the data and before you resume running the program, or there is an ENTER↑ in the program immediately after the R/S instruction.

Changing One Instruction

Changing or correcting one step of your program is easy with your HP-25 calculator because of the features built into it. Once the error has been found, use SST or BST in PRGM mode or GTO in RUN mode to display the step *preceding* the step to be changed. For example, to change the instruction in step 06, you need to display step 05. If you wish to change the step, simply press the correct key or keys for step 06. They will write over and replace the incorrect information already stored in that step.

If step 06 is an extra step in your program, press g NOP (*no operation*). This instruction tells the calculator not to perform any operation here.

Example Program. The program represented below is designed to take the cube root of a number.

Press	Display
ENTER↑	01 31
3	02 03
g 1/x	03 15 22
f y^x	04 14 03

Suppose that upon reviewing the program with the SST key, however, you discover you have keyed in the following mistake-ridden program:

Press	Display	
ENTER↑	01 31	
3	02 03	
g %	03 15 21	Oops! You pressed the wrong key.
x⇄y	04 21	And you pressed it again by mistake.
f y^x	05 14 03	

Set the PRGM-RUN switch to PRGM PRGM ▯▮ RUN, press [f] [PRGM], and key in this mistake-ridden second program now.

To correct the program, press [BST] three times to display step 02. Then correct the first mistake by keying in the correct keys for step 03.

Press	Display	
	02 03	First display this step.
[g] [1/x]	03 15 22	Then press the correct keys for step 03.

With step 03 displayed you are ready now to correct step 04. Since this is an unwanted extra step, use the [g] [NOP] function to replace its contents.

Press	Display	
	03 15 22	Display step 03 to correct step 04.
[g] [NOP]	04 15 74	Press [g] [NOP] so that the calculator will not perform an operation here.

Now set the PRGM-RUN switch back to RUN PRGM ▮▯ RUN and press [f] [PRGM] to reset the calculator to step 00. The example below will help you determine whether or not you have corrected the program.

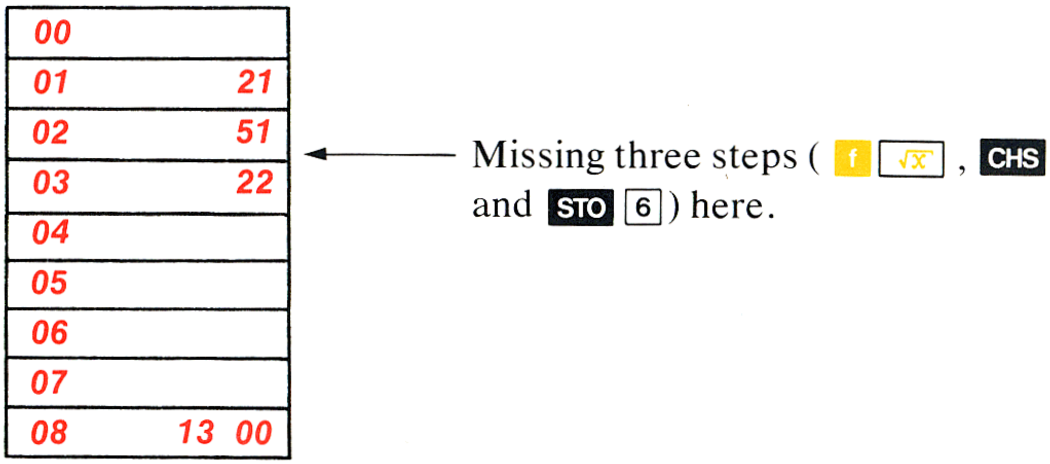
Example. Find the cube root of 8 and then of 125.

Press	Display
8 [R/S]	2.00
125 [R/S]	5.00

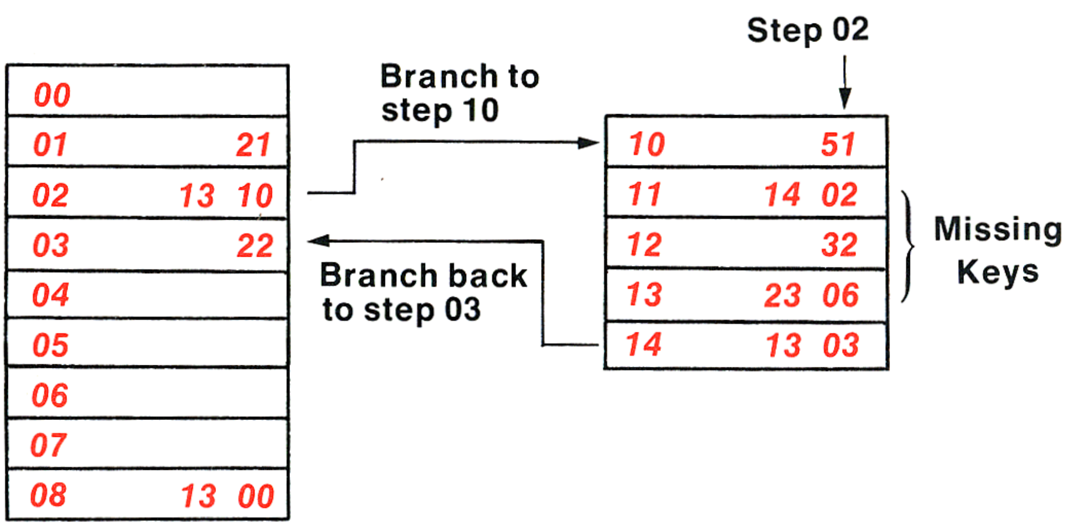
Adding Instructions

If you have recorded a medium-sized program and have left out a crucial sequence of keystrokes right in the middle, you do not have to start over. The missing sequence of keystrokes can be recorded in the available steps following your program. You can then use the [GTO] key to make an unconditional branch to the sequence when it is needed and then make a second unconditional branch back to the main part of your program at the end of the sequence.

The program segment shown below should make this more clear. Three keys are missing between steps 02 and 03.



In order to add the missing steps we need to branch to one of the available program steps in program memory. The corrected program is shown below.



Notice in particular that the instruction originally stored in step 02 is now stored in step 10. Step 02 now contains an unconditional branch instruction to step 10. The missing keys are stored in steps 11 through 13 and the instruction stored in step 14 is an unconditional branch back to step 03 in the main program.

Program Applications

The following two programs are provided as additional examples to test your programming skills. Only the purpose of each program is explained. See if you can figure out how each program works on your own.

Factorial

This program calculates the factorial of an input value "n" $[n(n\text{-}1)(n\text{-}2) \ldots 3 \times 2 \times 1]$. (For the special case where $n = 0$, $0! = 1$.) Switch to PRGM mode PRGM ▮▮ RUN and press f PRGM before keying in the following list of keys.

Keys	Display
g ABS	01 15 03
f INT	02 14 01
STO 1	03 23 01
g x=0	04 15 71
GTO 1 4	05 13 14
1	06 01
f x=y	07 14 71
GTO 1 6	08 13 16
x⇄y	09 21
1	10 01
−	11 41
STO × 1	12 23 61 01
GTO 0 6	13 13 06
1	14 01
GTO 0 0	15 13 00
RCL 1	16 24 01

Now switch back to RUN mode PRGM ▮▮ RUN and press f PRGM so you can try the following example.

Example. Calculate the number of ways six people can line up for a photograph.

Method: $P_6 = 6!$

Press	Display	
6 R/S	720.00	$(6 \times 5 \times 4 \times 3 \times 2 \times 1)$

Converging Series

This program uses the following series to approximate the value of "*e*" ($e = 1/0! + 1/1! + 1/2! + \ldots + 1/n!$). It then tests each approximation against the value for "*e*" generated by the calculator by pressing [1] [g] [e^x]. Each approximation is displayed, then the difference between the approximation and the calculator's value for "*e*" is displayed. When the two values are equal, the program stops and displays the number of terms it took for the series to converge.

Switch to PRGM mode PRGM [switch] RUN and press [f] [PRGM] before keying in the following list of keys.

Keys	Display
[1]	01 01
[STO] [0]	02 23 00
[STO] [1]	03 23 01
[ENTER↑]	04 31
[g] [1/x]	05 15 22
[RCL] [0]	06 24 00
[+]	07 51
[STO] [0]	08 23 00
[1]	09 01
[g] [e^x]	10 15 07
[f] [x=y]	11 14 71
[GTO] [2] [6]	12 13 26
[x⇄y]	13 21
[f] [FIX] [9]	14 14 11 09
[f] [PAUSE]	15 14 74
[f] [PAUSE]	16 14 74
[−]	17 41
[f] [PAUSE]	18 14 74
[R↓]	19 22
[RCL] [1]	20 24 01
[1]	21 01
[+]	22 51

STO 1	23 23 01
×	24 61
GTO 0 4	25 13 04
RCL 1	26 24 01
f FIX 2	27 14 11 02

Switch back to RUN mode PRGM ▮▮▮ RUN and press f PRGM before trying the program yourself by pressing R/S .

Your series should converge after 11.00 terms.

Afterword

If you have worked completely through this handbook, you should have a very good knowledge of all of the basic functions of the HP-25. But in fact you've only begun to see the power of the calculator. You'll come to understand it better and appreciate it more as you use the HP-25 daily to solve even the most complex mathematical expressions. At your fingertips you have a tool that was unavailable to Archimedes, Galileo, or Einstein. The only limits to the flexibility of the HP-25 are the limits of your own mind.