HEWLETT hp PACKARD

HP-65

STANDARD PAC

JUNE, 1974

CONTENTS

INTRODUCTION

The HP-65 Standard Pac is your introduction to using programs which are prerecorded on magnetic cards. Knowledge of programming the HP-65 is not required to use the pac, however, familiarity with the introduction of the Owner's Handbook will aid your understanding.

The Standard Pac is comprised of programs which demonstrate the simplicity of operation, the versatility and the computational power of your HP-65 Programmable Pocket Calculator. It contains programs of interest to engineers, scientists, statisticians, navigators, surveyors, doctors, businessmen, and people in many other technical and professional fields. The pac is only intended to show a sampling in these areas.

For each program the Standard Pac provides a description, general user instructions, user instructions specifically for the example problem(s), a prerecorded magnetic card (in the plastic card case) and program listings (at the back of the Pac). There are also two diagnostic programs for checking calculator operations, a head cleaning card which is used occassionally to clean the HP-65 magnetic card read/write head, and twenty blank magnetic cards which may be used to record programs that you write. For your convenience 20 blank pocket instruction cards are included to hold your favorite programs.

Six programs in this pac are representative examples of programs in "Application Pacs" which may be purchased from Hewlett-Packard. These are *Mean, Standard Deviation, Standard Error* (Statistics Pac I); *Quadratic Equation* (Mathematics Pac I); *Integer Base Conversion* (Mathematics Pac II); *Body Surface Area* (Medical Pac I); *PI Network Impedance Matching* (Electrical Engineering Pac I) and *EDM Slope Reduction-Given Δ Elevation* (Surveying Pac I). Additional pacs in various fields will be made available as they are developed. Each application pac includes prerecorded magnetic cards, a card case, 20 blank pocket instruction cards, and an instruction booklet with program descriptions, formulas, example problems, user instructions and program listings.

Individual listings and documentation of all programs in HP-65 application pacs are available through the HP-65 Users' Library* which is a collection of programs from many disciplines submitted by HP-65 users. Through the Users' Library and application pac development, Hewlett-Packard hopes to provide HP-65 programs useful in a great variety of applications. Your inputs, both comments and Users' Library contributions, will be of great help to us in our endeavor to provide you with high quality programs for your HP-65 Calculator.

*Domestic U.S.A. only.

FORMAT OF USER INSTRUCTIONS

The completed User Instruction Form—which accompanies each program—is your guide to operating the programs in this Pac.

The form is composed of five labeled columns. Reading from left to right, the first column, labeled STEP, gives the instruction step number.

The INSTRUCTIONS column gives instructions and comments concerning the operations to be performed.

The INPUT-DATA/UNITS column specifies the input data, and the units of data if applicable. Data input keys consist of [0] to [9] and decimal point (the numeric keys), [EEX] (enter exponent), and [CHS] (change sign).

The KEYS column specifies the keys to be pressed after keying in the corresponding input data. Where the [ENTER↑] key is used, it is indicated by [↑]. All other key designations are identical to those appearing on the HP-65. Ignore any blank spaces in the KEYS columns.

The OUTPUT-DATA/UNITS column specifies intermediate and final outputs and their units where applicable.

The following illustrates the User Instruction Form for Program STD–13A, *Reconcile Checking Account.*

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Initialize		D		0.00
3	Input statement balance	SB	A		SB
4	Repeat 4 for out checks	$C_1 \ldots C_n$	B		$C_1 \ldots C_n$
5	Repeat 5 for each outstanding				
	deposit	$D_1 \ldots D_m$	C		$D_1 \ldots D_m$
6	Compute final balance		E	A	FB
7	Recall statement balance		R/S		SB
	and/or sum of out checks		E	B	ΣC_i
	and number of checks		R/S		n_c
	and/or sum of out deposit		E	C	ΣD_i
	and number of deposits		R/S		m_D
8	To add additional checks go to				
	4, to add additional deposits go				
	to 5.				
9	For new case go to 2.				

STEP 1: Step 1 of the example is "Enter program". This calls for the entry of the prerecorded magnetic card into the HP-65 (See *Entering a Program,* on page 7).
STEP 2: This step "initializes" or prepares the calculator for proper program execution. Pressing the **D** key would perform the initialization in this case.

STEP 3: This step stores the statement balance. Press the applicable data input keys then press **A**. The statement balance is displayed after program execution ends.

STEP 4: This step is a repetition instruction as signified by the bold border enclosing the instructions. To perform Step 4, press the applicable data input keys to input the first outstanding check value. Press **B** to initiate program execution. The check value is still displayed after execution ends. Repeat the procedure for all outstanding checks. When all outstanding checks have been input, go to Step 5.

STEP 5: This step is also a repetition instruction. Outstanding deposits are the inputs. To perform Step 5 press applicable data input keys and press **C**. Repeat the procedure for all outstanding deposits. When all outstanding deposits have been input, go to Step 6.

STEP 6: This step computes the final balance. Press **E**, then **A** to display the final balance.

STEP 7: This step recalls values. Press **R/S** to display bank statement balance. Press **E**, then **B** to compute and display the sum of checks outstanding. Press **R/S** to display the number of checks outstanding. Press **E**, then **C** to compute and display the sum of deposits outstanding. Press **R/S** to display the number of deposits outstanding.

STEP 8: This step provides for the inclusion of additional checks and deposits outstanding: for checks start at Step 4, for deposits start at Step 5.

STEP 9: This step gives instructions on starting a new case. In this program, go to Step 2 and initialize.

In addition to the General User Instruction Form, each example problem in the Standard Pac is accompanied by Example User Instructions. The Example User Instructions differ from the General User Instructions in that they include numeric values instead of variable names. Also, each step number in the Example User Instructions corresponds to the step executed in the General User Instructions. Example User Instructions are included in the Standard Pac to help you become familiar with the HP-65 and the use of prerecorded magnetic cards. They are not generally included in HP-65 Application Pacs.

ENTERING A PROGRAM

From the card case supplied with this application pac, select a program card.

Set W/PRGM-RUN switch to RUN.

Turn the calculator ON. You should see 0.00

Gently insert the card (printed side up) in the right, lower slot as shown. When the card is part way in, the motor engages it and passes it out the left side of the calculator. Sometimes the motor engages but does not pull the card in. If this happens, push the card a little farther into the machine. Do not impede or force the card; let it move freely. (The display will flash if the card reads improperly. In this case, press [CLX] and reinsert the card.)

When the motor stops, remove the card from the left side of the calculator and insert it in the upper "window slot" on the right side of the calculator.

The program is now stored in the calculator. It remains stored until another program is entered or the calculator is turned off.

DAY OF THE WEEK

This program computes the day of the week for any date since September 14, 1752.*

The calculator displays the day of the week as an integer from 0 to 6 with the following correspondence:

Integer Displayed	Day
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Formulas:

The day of the week is given by

$$\text{Day} = (n_1 + n_2 - n_3 + n_4 + D - 1) - 7 \times \text{Int}\left[\frac{(n_1 + n_2 - n_3 + n_4 + D - 1)}{7}\right]$$

where:

$n_1 = \text{Int}\,(13(M' + 1)/5)$

$n_2 = \text{Int}\,(5Y'/4)$

$n_3 = \text{Int}\,(Y'/100)$

$n_4 = \text{Int}\,(Y'/400)$

D = Day of the Month

$Y' = Y - \text{Int}\left(.6 + \frac{1}{M}\right)$

*September 14, 1752 is the date that England and its colonies switched from the Julian Calendar to the current Gregorian Calendar. At that time eleven days were suppressed interrupting the continuity of the calendar. The suppression was necessary since the Julian Calendar was falling behind by 3 days every 400 years. This problem is corrected in the Gregorian Calendar by eliminating 3 leap years every 400 years. In the Gregorian system, century years (1700, 1800, 1900, etc.) are not leap years unless they are divisible by 400 (2000, 2400, 2800, etc.)

$$M' = M + 12\left[\text{Int}\left(.6 + \frac{1}{M}\right)\right]$$

Y = Year

M = Month of the year

(Int is the integer function of the HP-65)

Note:

The program has no checks for invalid inputs.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Input a date:				
	Month (January=1 and				
	December=12)	M	A		M
	and Day of the Month	D	B		D
	and Year	Y	C		Y
3	Calculate Day of the Week		D		Day
4	For new date go to Step 2 and				
	change any or all of the values				

Examples:

1. Von Ohain made the first jet powered flight on August 27, 1939. What was the day of the week?

   Answer: 0.00 (Sunday)

EXAMPLE 1 USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Day of the Week;				
	as shown on page 7 of this				
	manual)				
2	Input the Date				
	Month	8	A		8.00
	and Day of the Month	27	B		27.00
	and Year	1939	C		1939.00
3	Compute Day of the Week		D		0.00

2. After completing 492 deep soundings, taking 263 water temperature observations, dredging for 133 bottom samples, trawling 151 times and covering 68,890 nautical miles, the H.M.S. Challenger returned to England on May 24, 1876. The most important voyage in the history of oceanography was over. What was the day of the week?

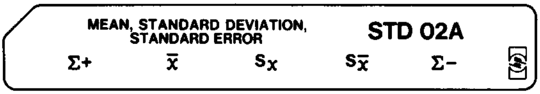
Answer: 3.00 (Wednesday)

EXAMPLE 2 USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Day of the				
	Week; as shown on page 7 of				
	this manual)				
2	Input the Date				
	Month	5	A		5.00
	and Day of the Month	24	B		24.00
	and Year	1876	C		1876.00
3	Compute Day of the Week		D		3.00

Notes

MEAN, STANDARD DEVIATION, STANDARD ERROR



Given a set of data points:

$$\{x_1, x_2, x_3, \ldots\ldots, x_n\}$$

the program calculates the mean, the standard deviation (by either of two methods) and the standard error of the mean (by either of two methods).

Formulas:

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

Standard deviation

$$s_x = \sqrt{\frac{\Sigma x_i^2 - n\bar{x}^2}{n-1}}$$

Alternate method

$$s_x' = \sqrt{\frac{\Sigma x_i^2 - n\bar{x}^2}{n}}$$

Standard error of the mean

$$s_{\bar{x}} = \frac{s_x}{\sqrt{n}}$$

Alternate method

$$s_{\bar{x}}' = \frac{s_x'}{\sqrt{n}}$$

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Initialize		RTN	R/S	0.00
3	Repeat 3 for each x	$x_1 \ldots x_n$	A		1...n
	(Eliminate any x values entered				
	in error—x error)	x error	E		
4	Compute $\bar{x}$		B		$\bar{x}$
5	Compute s_x and		C		s_x
6	s_x'		R/S		s_x'
7	Compute $s_{\bar{x}}$ and		D		$s_{\bar{x}}$
8	$s_{\bar{x}}'$		R/S		$s_{\bar{x}}'$
9	For a new case, go to 2				

Example:

In a recent survey to determine the average age of the wealthiest people in the U.S., the following data were obtained:

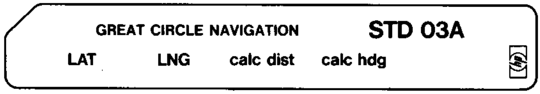
62 84 47 58 68 60 62 59 71 73

Of the ages given what is the mean, the standard deviation, and the standard error of the mean? Simply follow the keystep instructions to obtain the answers.

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Mean, Standard		[ ]	[ ]	
	Deviation; as shown on page 7		[ ]	[ ]	
	of this manual)		[ ]	[ ]	
2	Initialize		RTN	R/S	0.00
3	Input ages	62	A	[ ]	1.00
		84	A	[ ]	2.00
		47	A	[ ]	3.00
		58	A	[ ]	4.00
		68	A	[ ]	5.00
		60	A	[ ]	6.00
		62	A	[ ]	7.00
		59	A	[ ]	8.00
		71	A	[ ]	9.00
		73	A	[ ]	10.00
4	Compute $\bar{x}$		B	[ ]	64.40
5	Compute s_x		C	[ ]	10.10
6	Compute s_x'		R/S	[ ]	9.58
7	Compute $s_{\bar{x}}$		D	[ ]	3.19
8	Compute $s_{\bar{x}}'$		R/S	[ ]	3.03

GREAT CIRCLE NAVIGATION



This program accepts the coordinates of two points on the globe and calculates the great circle distance between them as well as the initial heading.

The program inputs are latitude and longitude of the source (LAT_S, LNG_S) and latitude and longitude of the destination (LAT_D, LNG_D), (the above are expressed in the notation degrees $\boxed{\cdot}$ minutes, i.e., 15.30 means 15° 30′).

Northern latitudes are entered as positive values while southern ones are entered as negative values.

Western longitudes are entered as positive values while eastern ones are entered as negative values.

The outputs are great circle distance (Dist) in nautical miles and initial great circle heading (Hdg) in decimal degrees. (You may convert back to degrees, minutes, seconds by pressing $\boxed{f}$ $\boxed{\to D.MS}$.)

Any number of consecutive legs may be linked together without any reentry of data. Short legs are recommended since only the initial course heading is given and intermediate heading changes may well be desirable.

Note:

No leg should pass more than half way around the earth. Legs directly north or south may cause flashing zeros when heading is calculated.

Formulas:

$$\begin{aligned} \text{Dist} = \cos^{-1} & [\sin(LAT_S)\sin(LAT_D) + \cos(LAT_S) \\ & \cos(LAT_D)\cos(LNG_D - LNG_S)] \times 60 \end{aligned}$$

$$\text{Hdg} = \cos^{-1}\left[\frac{\sin(LAT_D) - \cos(\text{Dist}/60)\sin(LAT_S)}{\sin(\text{Dist}/60)\cdot\cos(LAT_S)}\right]$$

If $\sin(LNG_S - LNG_D) < 0$ then $\text{Hdg} = 360 - \text{Hdg}$

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program		[ ] [ ]	
2	Initialize		[RTN] [R/S]	0.00
3	Input starting latitude	(deg . min)	[A] [ ]	(dec. deg.)
	and input starting longitude	(deg . min)	[B] [ ]	(dec. deg.)
4	Input destination latitude	(deg . min)	[A] [ ]	(dec. deg.)
	and input destination longitude	(deg . min)	[B] [ ]	(dec. deg.)
5	Calc. Great Circle Dist		[C] [ ]	(naut. miles)
	and/or calc. initial heading		[D] [ ]	(dec. deg.)
6	Go to step 4 to calculate		[ ] [ ]	
	next leg		[ ] [ ]	
7	To reinitialize go to step 2.		[ ] [ ]	

Example:

A navigator wishes to follow great circle courses from Chicago to St. Louis to New Orleans. Find the great circle distances and initial courses.

	LAT	LNG
Chicago	41° 50′ N	87° 36′ W
St. Louis	38° 38′ N	90° 12′ W
New Orleans	29° 56′ N	90° 04′ W

Note:

After entries, the angle in decimal degrees is displayed.

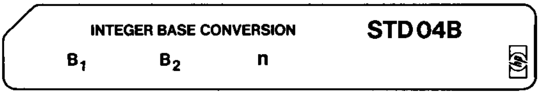
Answers: $Dist_1$ = 225.91 Nautical miles Hdg_1 = 212.66°

$Dist_2$ = 522.04 Nautical miles Hdg_2 = 179.24°

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Great Circle				
	Navigation; as shown on page 7				
	of this manual)				
2	Initialize		RTN	R/S	0.00
3	Input LAT Chicago	41.50	A		41.83
	Input LNG Chicago	87.36	B		87.60
4	Input LAT St. Louis	38.38	A		38.63
	Input LNG St. Louis	90.12	B		90.20
5	Calc. Great Circle Distance		C		225.91
	Calc. initial heading		D		212.66
4	Input LAT New Orleans	29.56	A		29.93
	Input LNG New Orleans	90.04	B		90.07
5	Calc. Great Circle Distance		C		522.04
	Calc. initial heading		D		179.24

INTEGER BASE CONVERSION



This program can be used to convert an integer n in base B_1 to an equivalent integer in base B_2, where B_1, B_2 are integers such that $2 \leqslant B_i \leqslant 10$ (i = 1, 2).

n is first converted to a decimal integer then the decimal integer is converted to an integer in base B_2.

Note: A non-integer entry is truncated to an integer which then is converted to its equivalent integer in the specified base.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program			
2	Input the base B_1 of n	B_1	A	B_1
	and the desired base B_2	B_2	B	B_2
3	Input n in B_1 and convert to B_2	$n(B_1)$	C	$n(B_2)$
4	For new n go to step 3			
	For new base go to step 2			

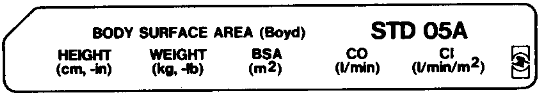
Example:

$110_2 = 6_8$

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Integer Base			
	Conversion; as shown on page			
	7 of this manual.)			
2	Input B_1	2	A	2.00
	and B_2	8	B	8.00
3	Input n and convert	110	C	6.00

BODY SURFACE AREA (BOYD)



This program calculates the body surface area in square meters from the patient's height (in inches or centimeters) and weight (in pounds or kilograms). If height is in inches, enter it as a negative number and it will be converted to centimeters. Enter centimeters as a positive number. If weight is in pounds, enter as a negative number, and it will be converted to kilograms. Enter kilograms as a positive number.

Also, the cardiac index (a normalized measure of cardiac blood output which removes the influence of patient size) is calculated from the height, weight, and cardiac output. The formula used for the body surface area calculation is that of Edith Boyd *(The Growth of the Surface Area of the Human Body,* U. of Minn. Press, 1935), and is claimed to be valid throughout the life span.

Formulas:

$$BSA = (3.20W^{0.7285 - .0188 \log W} H^{0.3}) \div 10^4$$

$$CI = CO/BSA$$

$$W = 1000\, W_t$$

BSA = Body surface area (square meters)

W = Body weight (grams)

W_t = Body weight (kilograms)

H = Height (centimeters)

CI = Cardiac index (liters/min/m^2)

CO = Cardiac output (liters/min)

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Input height (cm)	Ht (cm)	A		Ht (cm)
	or input height (in.)	Ht (in)	CHS	A	Ht (cm)
3	Input weight (kg)	Wt (kg)	B		Wt (kg)
	or input weight (lb.)	Wt (lb)	CHS	B	Wt (kg)
4	Calculate BSA		C		BSA (m^2)
5	Input cardiac output	CO (l/min.)	D		CO(l/min)
6	Calculate cardiac index		E		CI(l/min/m^2)
7	For new case go to 2				

Example:

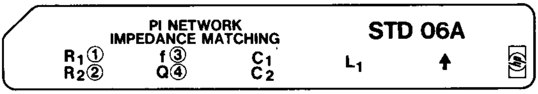
Height = 70 inches, weight = 170 pounds, cardiac output = 8 liters/minute. Height is entered as (–70) and weight is entered as (–170), giving:

Body Surface Area (BSA) = 1.96 square meters
Cardiac Index = 4.08 liters/minute/square meter

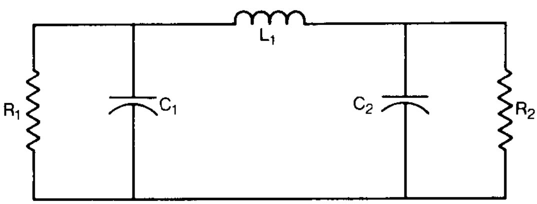
EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Body Surface				
	Area; as shown on page 7 of				
	this manual)				
2	Input height in inches	70	CHS	A	177.80
3	Input weight in lbs.	170	CHS	B	77.27
4	Calculate BSA		C		1.96
5	Input cardiac output	8	D		8.00
6	Calculate cardiac index		E		4.08

PI NETWORK IMPEDANCE MATCHING



A lossless network is often used to match between two resistive impedances, R_1 and R_2, as shown.



Given the values of R_1 and R_2, the frequency (f), and the desired circuit Q (center frequency/desired half-power bandwidth), the values of C_1, C_2, and L_1 are found from the following formulas.

Formulas:

$$X_{C1} = \frac{R_1}{Q} \qquad C_1 = \frac{1}{2\pi f X_{C1}}$$

$$X_{C2} = \frac{R_2}{\left[\frac{R_2}{R_1}(Q^2+1) - 1\right]^{1/2}} \qquad C_2 = \frac{1}{2\pi f X_{C2}}$$

$$X_{L1} = \frac{QR_1}{Q^2+1}\left[1 + \frac{R_2}{QX_{C2}}\right] \qquad L_1 = \frac{X_{L1}}{2\pi f}$$

Notes:

1. R_1 must always be greater than R_2 and

$$Q > \sqrt{R_1/R_2 - 1}$$

2. Circled numbers on the magnetic card designate the register in which a variable is stored.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program		[ ]	[ ]	
2	Initialize		[RTN]	[R/S]	0.0000×10^0
3	Input R_1	R_1 (ohms)	[E]	[A]	R_1 (ohms)
	and R_2	R_2 (ohms)	[A]	[ ]	R_2 (ohms)
	and f	f(Hz)	[E]	[B]	f(Hz)
	and Q	Q	[B]	[ ]	Q
4	Compute C_1		[E]	[C]	C_1 (farads)
5	Compute C_2		[C]	[ ]	C_2 (farads)
6	Compute L_1		[D]	[ ]	L_1 (henrys)
7	Recall inputs (optional)		[ ]	[ ]	
	R_1		[RCL]	[1]	R_1 (ohms)
	and/or R_2		[RCL]	[2]	R_2 (ohms)
	and/or f		[RCL]	[3]	f (Hz)
	and/or Q		[RCL]	[4]	Q
8	For new case change appropriate		[ ]	[ ]	
	input in step 3.		[ ]	[ ]	

Example:

$R_1 = 500 \quad R_2 = 50 \quad Q = 10 \quad 4 \times 10^6$ (4MHz)

Calculate $C_1 = 7.9577 \times 10^{-10} \approx 796$ pF

$C_2 = 2.4006 \times 10^{-9} \approx 2400$ pF

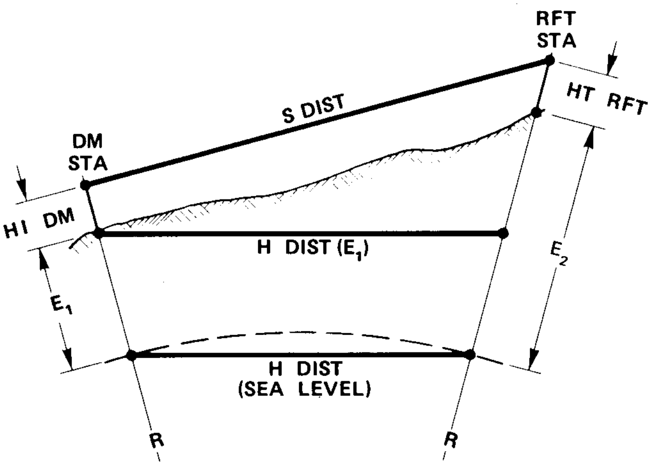
$L_1 = 2.5639 \times 10^{-6} \approx 2.56\ \mu$H

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
	(PI Network Impedance				
	Matching; as shown on page 7				
	of this manual)				
2	Initialize		RTN	R/S	0.0000×10^0
3	Input R_1	500	E	A	5.0000×10^2
	and R_2	50	A		5.0000×10^1
	and f	4×10^6	E	B	4.0000×10^6
	and Q	10	B		1.0000×10^1
4	Compute C_1		E	C	7.9577×10^{-10}
5	Compute C_2		C		2.4006×10^{-9}
6	Compute L_1		D		2.5639×10^{-6}
7	Recall inputs (optional)				
	R_1		RCL	1	5.0000×10^2
	and/or R_2		RCL	2	5.0000×10^1
	and/or f		RCL	3	4.0000×10^6
	and/or Q		RCL	4	1.0000×10^1

EDM SLOPE REDUCTION - GIVEN Δ ELEVATION

Taking into consideration the curvature of the earth, this program reduces slope distance to horizontal distance at the instrument station elevation. The program assumes the slope distance between two points having known elevations was measured using an electronic distance measuring instrument. As options, the program will reduce the slope distance to a horizontal distance at sea level, and to a horizontal distance at any specified elevation. The value used for the radius of the earth is 20,906,000 feet or 6,378,200 meters.



$$H\ Dist = \left[\sqrt{\frac{(S\ Dist)^2 - (E_2 + HT\ Rft - E_1 - HI\ DM)^2}{(R + E_1 + HI\ DM)(R + E_2 + HT\ Rft)}}\right]\left[R + E\right]$$

where:

S Dist = Slope Distance
E_1 = Elevation of Instrument Station
HI DM = Height of Instrument
E_2 = Elevation of Reflector Station
Ht Rft = Height of Reflector
R = Radius of the Earth (20,906,000 ft.)
E = Elevation of Horizontal Distance
H Dist = Horizontal Distance
E_S = Specified Elevation

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Input slope distance in feet	S Dist(ft)	A		S Dist(ft)
	or in meters	S Dist(m)	B		S Dist(m)
3	Input height of distance meter	HI DM	C		HI DM
4	Input height of reflector	HT Rft	C		HT Rft
5	Input elevation at DM station	E_1	D		E_1
6	Input elevation at Rft station	E_2	D		E_2
7	Optional: Input specified				
	elevation	E_s	D		E_s
8	Compute horizontal distance		E		H Dist(E_1)
9	Optional: Compute H Dist at				
	sea level		R/S		H Dist(SL)
10	Optional: Compute H Dist at				
	E_s		R/S		H Dist (E_s)

Example:

Slope Distance	S Dist = 10,000 ft.
Height of DM	HI DM = 5.12
Height of Reflector	HT Rft = 4.75
Elev at DM Station	E_1 = 1000.00
Elev at Reflector Station	E_2 = 3590.63
Specified Elevation	E_S = 2000
Horizontal Distance (E_1)	H Dist = 9657.83
Horizontal Distance (Sea Level)	H Dist = 9657.37
Horizontal Distance (E_S)	H Dist = 9658.30

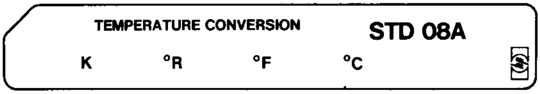
EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (EDM Slope				
	Reduction–Given Δ Elevation;				
	as shown on page 7 of this				
	manual)				
2	Input slope distance in feet	10000	A		10000
3	Input height of distance meter	5.12	C		5.12
4	Input height of reflector	5.75	C		5.75
5	Input elevation at DM station	1000	D		1000
6	Input elevation at Rft station	3590.63	D		3590.63
7	Input specified elevation	2000	D		2000
8	Compute horizontal distance		E		9657.83
9	Compute H Dist at sea level		R/S		9657.37
10	Compute H Dist at specified				
	elevation		R/S		9658.30

Note:

Programs in Surveyor Pac I are generally not compatible with metric units.

TEMPERATURE CONVERSION



This program converts temperature interchangeably between degrees Celsius (Centigrade), Kelvin, Fahrenheit and Rankine. The following standard relationships are used in the conversions:

$$°C = \frac{5}{9}\ (°F - 32)$$

$$°F = °R - 459.67$$

$$°R = \frac{9}{5}\ K$$

$$K = °C + 273.15$$

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Initialize		RTN	R/S	
3	Input either				
	K	K	A		0.00*
	or °R	°R	B		0.00*
	or °F	°F	C		0.00*
	or °C	°C	D		0.00*
4	Then convert to either				
	K		A		K
	or °R		B		°R
	or °F		C		°F
	or °C		D		°C
5	For new case go to 3				
	*Note: If display is not zero				
	go to 2.				

Example:

Convert 212°F to K

Answer: 373.15K

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Temperature		[ ] [ ]	
	Conversion; as shown on page 7		[ ] [ ]	
	of this manual)		[ ] [ ]	
2	Initialize		[RTN] [R/S]	
3	Input °F	212°F	[C] [ ]	0.00
4	Convert to K		[A] [ ]	373.15 K

WEIGHT-MASS CONVERSION

This program converts weight and/or mass interchangeably between pounds, ounces, kilograms, grams and slugs. Combinations of the following equalities are used to perform conversions:

1 pound = 16 ounces
1 ounce = 28.349523 grams
1 kilogram = 1000 grams
1 slug = 32.174 pounds mass

Note:

Zero is an invalid input.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program			
2	Input either			
	pounds	lbs	A	0.00
	or ounces	oz	B	0.00
	or kilograms	kg	C	0.00
	or grams	gm	D	0.00
	or slugs	slugs	E	0.00
3	Then convert to either			
	pounds		A	lbs
	or ounces		B	oz
	or kilograms		C	kg
	or grams		D	gm
	or slugs		E	slugs
4	For new case go to 2.			

Example:

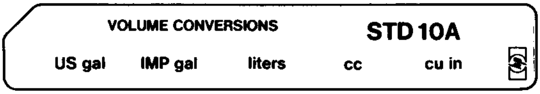
Convert 10 pounds to kilograms

Answer: 4.54 kg

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Weight–Mass		[ ] [ ]	
	Conversion; as shown on page		[ ] [ ]	
	7 of this manual)		[ ] [ ]	
2	Input pounds	10	[A] [ ]	0.00
3	Convert to kilograms		[C] [ ]	4.54

VOLUME CONVERSIONS



This program converts volume interchangeably between U.S. gallons, imperial gallons, liters, cubic centimeters and cubic inches. Combinations of the following equalities are used to perform conversions:

1 U.S. gallon = 3.7854 liters
1 Imperial gallon = 1.20095 U.S. gallons
1 liter = 1000 cubic centimeters
1 cubic inch = 16.387064 cubic centimeters

Note:

Zero is an invalid input.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program			
2	Input either			
	U.S. gallons	U.S. gal.	A	0.00
	or Imperial gallons	Imp. gal.	B	0.00
	or liters	l	C	0.00
	or cubic centimeters	cc	D	0.00
	or cubic inches	in^3	E	0.00
3	Then convert to either			
	U.S. gallons		A	U.S. gal.
	or Imperial gallons		B	Imp. gal.
	or liters		C	l
	or cubic centimeters		D	cc
	or cubic inches		E	in^3

Example:

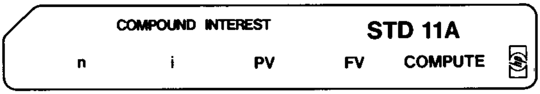
Convert 2400 cubic centimeters to cubic inches

Answer: 146.46 cubic inches

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Volume		[ ] [ ]	
	Conversions; as shown on page		[ ] [ ]	
	7 of this manual)		[ ] [ ]	
2	Input cubic centimeters	2400	[D] [ ]	0.00
3	Convert to cubic inches		[E] [ ]	146.46

COMPOUND INTEREST



This program computes the answers to compound interest problems using the formulas below:

1. $$n = \frac{\ln(FV/PV)}{\ln(1 + i/100)}$$

2. $$i = [\,(FV/PV)^{1/n} - 1\,] \times 100$$

3. $$PV = FV\,(1 + i/100)^{-n}$$

4. $$FV = PV\,(1 + i/100)^{n}$$

Note: Formulas 1, 2 and 3 are derived from 4

where:

n = Number of time (compounding) periods.

i = Interest rate per time period (in percent).

PV = Present value (value at the beginning of the first time period).

FV = Future value (value at the end of n time periods).

Any three of the variables (n, i, PV, FV) can be inputs. The program computes and stores the fourth variable. Input variables can be entered in any order and need not all be reentered to change one variable.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program		[ ] [ ]	
2	Initialize		[RTN] [R/S]	
3	Input 3 of the following		[ ] [ ]	
	n	n	[A] [ ]	n
	and/or i	i(%)	[B] [ ]	i(%)
	and/or PV	PV	[C] [ ]	PV
	and/or FV	FV	[D] [ ]	FV
4	Compute the remaining variable		[ ] [ ]	
	n		[E] [A]	n
	or i		[E] [B]	i(%)
	or PV		[E] [C]	PV
	or FV		[E] [D]	FV
5	To modify problem go to 3		[ ] [ ]	
	and change appropriate input(s).		[ ] [ ]	

Example:

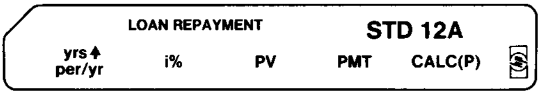
What amount must be invested today to have $15,000 at the end of 20 years if the interest rate is 7% compounded quarterly?

Answer: $3744.02

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Compound		[ ] [ ]	
	Interest; as shown on page 7		[ ] [ ]	
	of this manual)		[ ] [ ]	
2	Initialize		[RTN] [R/S]	0.00
3	Input		[ ] [ ]	
	n (n = 20 x 4)	80	[A] [ ]	80.00
	and i (i = 7 ÷ 4)	1.75	[B] [ ]	1.75
	and FV	15000	[D] [ ]	15000.00
4	Compute PV		[E] [C]	3744.02

LOAN REPAYMENT



Given the term for a direct reduction loan in years, the number of periodic payments per year, and the annual interest rate in percent this program computes:

1. Periodic payment amount, if the principal value borrowed is given.

or

2. The principal value if the periodic payment is given.

A one half cent round-off routine is incorporated so that the answers are correct to the nearest cent.

Formulas:

$$PV = PMT \left[\frac{\left(1 + \frac{i}{100n}\right)^{yn} - 1}{\frac{i}{100n}\left(1 + \frac{i}{100n}\right)^{yn}} \right]$$

where:

PV = Principal value
PMT = Periodic payment
i = Annual interest rate in percent
y = Number of years
n = Number of payment periods per year

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Input number of years	y	↑		y
3	Input payment periods/year	n	A		yn
4	Input annual interest	i(%)	B		100
5	Input either principal value	PV	C		PV
	or payment	PMT	D		PMT
6	Calculate either				
	payment		E	D	PMT
	or principal value		E	C	PV
7	Perform steps 2 and 3, or step 4				
	or step 5 for new values.				
	Note: Unrounded values for				
	PMT or PV calculations are				
	stored in register 8 after				
	calculation.				

Example:

Find the quarterly payment for a thirty year mortgage at 8.75% with principal amount of $37,500.

Answer: $886.36

What principal value corresponds exactly to the periodic payment calculated?

Answer: 37,499.86

What would the payment be if the interest is 9.25%

Answer: $926.83

At the higher interest what would the monthly payments be?

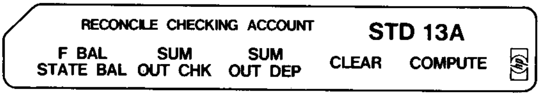
Answer: $308.50

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Loan Repayment				
	; as shown on page 7 of this				
	manual)				
2	Input number of years	30	↑		30.00
3	Input payment periods/year	4	A		120.00
4	Input annual interest	8.75	B		100.00
5	Input principal value	37500	C		37500.00
6	Calculate payment		E	D	886.36
5	Input payment		D		886.36
6	Calculate principal value		E	C	37499.86
	(the difference is due to				
	accumulated ½ cent roundoff)				
4	Input new interest	9.25	B		100.00
6	Calculate payment		E	D	926.83
2	Input number of years	30	↑		
3	Input payment periods/year	12	A		360.00
6	Calculate payment		E	D	308.50

Notes

RECONCILE CHECKING ACCOUNT



This program serves as an aid in reconciling personal checkbook tallies to bank statements. Inputs are outstanding checks (check which have not cancelled), outstanding deposits (deposits after the statement closing date), and bank statement balance. Outputs are final balance (this should agree with the personal checkbook tally), sum and total number of outstanding checks, and sum and number of outstanding deposits. All statement service charges should be subtracted from the checkbook tally before reconciling.

Formula:

$$FB = SB + \sum_{i=1}^{m_D} D_i - \sum_{i=1}^{n_c} C_i$$

where:

- FB = Final Balance
- SB = Bank Statement Balance
- D_i = Outstanding Deposit Number i
- C_i = Outstanding Check Number i
- m_D = Number of Outstanding Deposits
- n_c = Number of Outstanding Checks

Expressed differently, the current checkbook balance (FB) equals the bank statement balance (SB) plus deposits made after the statement closing date

$$\sum_{i=1}^{m_D} D_i$$

minus checks not received at the bank before the closing date

$$\sum_{i=1}^{n_c} C_i .$$

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Initialize		D		0.00
3	Input statement balance	SB	A		SB
4	Repeat 4 for out checks	$C_1 \ldots C_n$	B		$C_1 \ldots C_n$
5	Repeat 5 for each outstanding				
	deposit	$D_1 \ldots D_m$	C		$D_1 \ldots D_m$
6	Compute final balance		E	A	FB
7	Recall statement balance		R/S		SB
	and/or sum of out checks		E	B	ΣC_i
	and number of checks		R/S		n_c
	and/or sum of out deposit		E	C	ΣD_i
	and number of deposits		R/S		m_D
8	To add additional checks go to				
	4, to add additional deposits go				
	to 5.				
9	For new case go to 2.				

Example:

SB = $432.96 (Bank Statement Balance)

Outstanding checks are as follows:

$47.82
$ 5.63
$25.00
$36.47
$96.02

Outstanding deposits are as follows:

$100.00
$256.03

What should be the checkbook current balance?

Answer: $578.05

What is the dollar total of outstanding check?

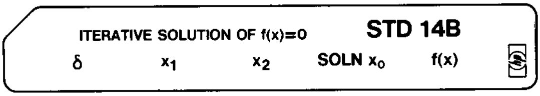
Answer: $210.94

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Reconcile				
	Checking Account; as shown on				
	page 7 of this manual)				
2	Initialize		D		0.00
3	Input statement balance	432.96	A		432.96
4	Input outstanding checks				
		47.82	B		47.82
		5.63	B		5.63
		25.00	B		25.00
		36.47	B		36.47
		96.02	B		96.02
5	Input outstanding deposits				
		100.00	C		100.00
		256.03	C		256.03
6	Compute final balance		E	A	578.05
7	Recall statement balance		R/S		432.96
	and/or sum of out checks		E	B	210.94
	and number of out checks		R/S		5.00
	and/or sum of out deposits		E	C	356.03
	and number of out deposits		R/S		2.00

Notes

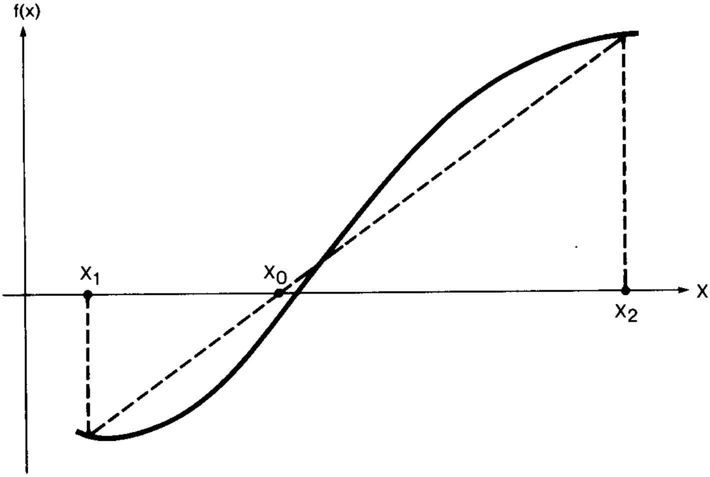
ITERATIVE SOLUTION OF f(x)=0



This program finds a solution x_0 of an equation $f(x) = 0$ by a modified regula falsi method. The user specifies the continuous real-valued function f(x), an accuracy tolerance δ, and two points x_1, x_2 such that $f(x_1) \cdot f(x_2) < 0$; that is, such that $f(x_1)$ and $f(x_2)$ are of opposite signs.

The basic formula used to obtain approximations to x_0 is

$$x_0 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)}\, f(x_2)$$

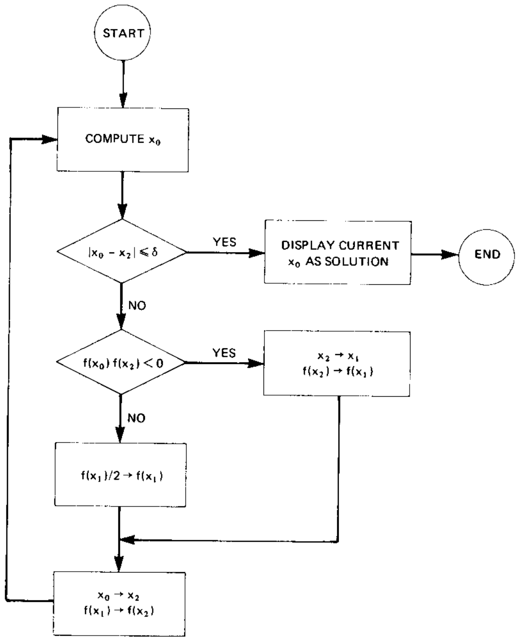


The estimates to be used for the next iteration are chosen according to the following rules such that the function values used at each iteration will always have opposite signs:

1. If $f(x_0) \cdot f(x_2) < 0$, replace $(x_1, f(x_1))$ by $(x_2, f(x_2))$.
   If $f(x_0) \cdot f(x_2) > 0$, replace $(x_1, f(x_1))$ by $\left(x_1, \frac{f(x_1)}{2}\right)$.
2. Replace $(x_2, f(x_2))$ by $(x_0, f(x_0))$.

The iterative process continues until $|x_0 - x_2| \leqslant \delta$.

Key in steps defining the function f(x) assuming the value x is in the X register. Thirty-three memory locations, the operational stack and registers R_7 and R_8 are available for f(x). (Register R_9 is also available for temporary storage only).



Notes:

1. If a function crosses the x axis more than once in the interval between x_1 and x_2, this program will find only one of the zero values. Also, if a function crosses the x axis an infinite number of times between x_1 and x_2, the iterative routine may not converge.

2. Once x_0 is obtained, the **E** key may be used to see if $f(x_0)$ is close enough to zero for the δ used. If not, a new δ may be chosen.

3. The **E** key can be used independently to evaluate f(x) for any x in the display.

Reference:

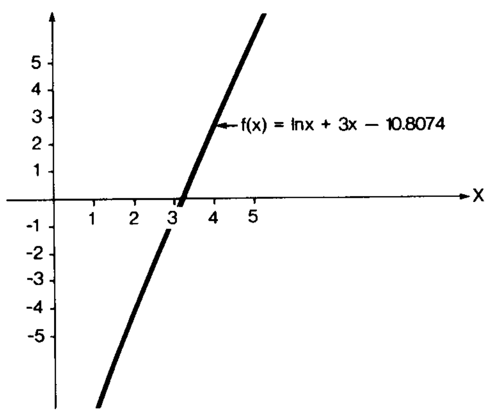
M. Dowell and P. Jarratt, "A Modified Regula Falsi Method for Computing the Root of an Equation," BIT 11, 1971, pp. 168–174.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program		[ ] [ ]	
2	Go to label E		[GTO] [E]	
3	Switch to W/PRGM mode		[ ] [ ]	15
4	Key in steps defining f(x)		[ ] [ ]	
	(followed by RTN)		[RTN] [ ]	24
5	Switch to RUN mode		[ ] [ ]	
6	Input desired δ	δ	[A] [ ]	δ
7	Input x_1	x_1	[B] [ ]	$f(x_1)$
8	Input x_2*	x_2	[C] [ ]	$f(x_2)$
	If $f(x_2)$ has the same sign as		[ ] [ ]	
	$f(x_1)$, choose new x_2 and		[ ] [ ]	
	repeat 8.		[ ] [ ]	
9	Compute solution x_0		[D] [ ]	x_0
10	Compute $f(x_0)$		[E] [ ]	$f(x_0)$
11	(Optional) input a new δ,	δ	[A] [ ]	δ
	then go to 9.		[ ] [ ]	
	(For a new f(x), go to 2.)		[ ] [ ]	
	*x_2 may be chosen smaller		[ ] [ ]	
	than x_1 if necessary.		[ ] [ ]	

Example

Find the solution of $\ln x+3x-10.8074 = 0$



$f(x) = \ln x + 3x - 10.8074$

Note:

Since the constant 10.8074 is a 6-digit number, it will be more efficient to store it in a register than to program it in the memory as program steps. We will store this constant in register R_7 before we key in the function f(x).

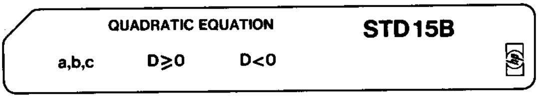
Answer: $x_0 = 3.21$

$f(x_0) = -4 \times 10^{-8}$

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Iterative				
	Solution of f(x) = 0; as shown				
	on page 7 of this manual)				
2	Store constant 10.8074 in R_7	10.8074	STO	7	10.81
	Go to label E		GTO	E	10.81
3	Switch to W/PRGM mode				15
4	Key in steps defining		f		31
	$f(x) = \ln x + 3x - 10.8074$		LN		07
			g	LST X	35 00
			3		03
			x		71
			+		61
			RCL	7	34 07
			–		51
			RTN		24
5	Switch to RUN mode				10.81
6	Input δ (say $\delta = 10^{-6}$)	10^{-6}	A		0.00
7	Input $x_1 = 1$	1	B		–7.81
8	Guess $x_2 = 2$	2	C		–4.11
	$f(x_2)$ has same sign as $f(x_1)$,	4	C		2.58
	guess $x_2 = 4$				
9	Compute solution x_0 (takes		D		3.21
	about 12 seconds)				
10	Compute $f(x_0)$		E		–0.00
	Change display format to see it		DSP	9	-4×10^{-8}

QUADRATIC EQUATION



The roots x_1, x_2 of $\quad ax^2 + bx + c = 0$

are given by $\quad x_{1,2} = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

However, better significance can be obtained in some cases by first computing the root with the largest absolute value using the following formula

$$x_1 = \frac{-ab}{|ab|}\left(\left|\frac{b}{2a}\right| + \sqrt{\frac{b^2 - 4ac}{4a^2}}\right)$$

then the smaller root by $\quad x_2 = \dfrac{c}{x_1\, a}$

If $\quad D = (b^2 - 4ac)/4a^2$

is positive or zero, the roots are real. Otherwise, they are complex, being

$$u \pm iv = \frac{-b}{2a} \pm \frac{\sqrt{4ac - b^2}}{2a}\, i$$

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Input coefficients				
	a	a	↑		a
	and b	b	↑		b
	and c	c	A		D
3	If D ⩾ 0 roots are real		B		root 1
			R/S		root 2
4	If D < 0 roots are complex		C		u
			R/S		v

Examples:

1. $2x^2 + 5x + 3 = 0$

   $(D = 0.06 > 0)$

   Answers: $x_1 = -1.50$

   $x_2 = -1.00$

EXAMPLE 1 USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Quadratic			
	Equation; as shown on page 7			
	of this manual)			
2	Input coefficients			
	a	2	↑	2.00
	and b	5	↑	5.00
	and c	3	A	0.06
3	Compute root 1		B	-1.50
	Compute root 2		R/S	-1.00

2. $2x^2 + 3x + 4 = 0$

   $(D = -1.44 < 0)$

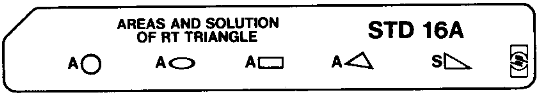
   Answers: $x_1 = -0.75 + 1.20i$

   $x_2 = -0.75 - 1.20i$

EXAMPLE 2 USER INSTRUCTIONS

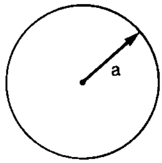
STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Quadratic			
	Equation; as shown on page 7			
	of this manual)			
2	Input coefficients			
	a	2	↑	2.00
	and b	3	↑	3.00
	and c	4	A	-1.44
4	Compute u		C	-0.75
	Compute v		R/S	1.20

AREAS AND SOLUTION OF RIGHT TRIANGLE



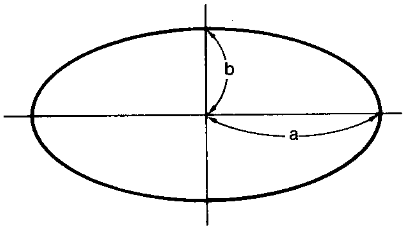
This program finds:

1. The area of a circle given the radius a



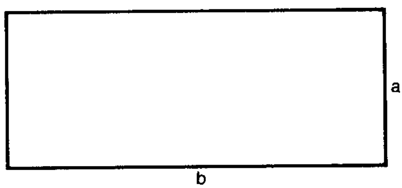
$$A = \pi a^2$$

2. The area of an ellipse given a and b



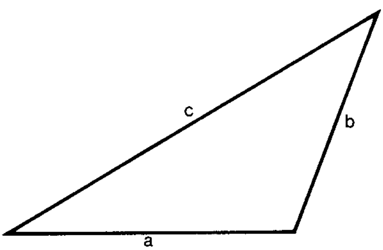
$$A = \pi ab$$

3. The area of a rectangle given a and b



$$A = ab$$

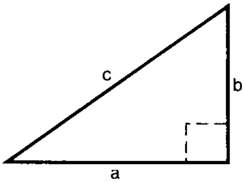
4. The area of a triangle given a, b, and c



$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \tfrac{1}{2}(a+b+c)$$

5. The third side of a right triangle given the other two. (If the hypotenuse is one of the known legs it must be entered as a negative value.)



$$c^2 = a^2 + b^2$$

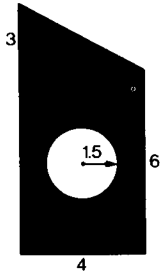
The program also keeps a running total of the areas it has computed. Areas computed using only positive inputs are added to this total. To subtract an area from this total make the last dimension entered negative. Using this feature it is possible to find the area of combinations of circles, ellipses, rectangle and triangles.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program		[ ] [ ]	
2	Initialize		[RTN] [R/S]	0.00
3	To find the area of a circle	± *a	[A] [ ]	Area
	or an ellipse	a	[↑] [ ]	
		± *b	[B] [ ]	Area
	or a rectangle	a	[↑] [ ]	
		± *b	[C] [ ]	Area
	or a triangle	a	[↑] [ ]	
		b	[↑] [ ]	
		± *c	[D] [ ]	Area
	or to find the third side of a		[ ] [ ]	
	right triangle	a	[↑] [ ]	
		b or –c	[E] [ ]	side 3
4	For next area go to 3.		[ ] [ ]	
5	For total of areas previously		[ ] [ ]	
	computed (this step must		[ ] [ ]	
	directly follow an area		[ ] [ ]	
	computation)		[R/S] [ ]	Σ Areas
6	To start new sum go to 2.		[ ] [ ]	
	*Note: Input as a negative		[ ] [ ]	
	number if the area is to be sub-		[ ] [ ]	
	tracted from the total being		[ ] [ ]	
	accumulated.		[ ] [ ]	

Example:

Find the area of this figure (the circle is to be subtracted from the total area).



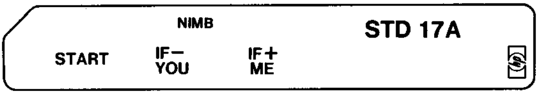
Answer: 22.93

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program (Areas and			
	Solution of Right Triangles; as			
	shown on page 7 of this manual)			
2	Initialize		RTN R/S	0.00
3–4	Compute third side of triangle			
	Input side	3	↑	3.00
	Input side	4	E	5.00
	Compute area of triangle			
	Input side	3	↑	3.00
	Input side	4	↑	4.00
	Input side	5	D	6.00
	Compute area of rectangle			
	Input side	6	↑	6.00
	Input side	4	C	24.00
	Delete area of circle			
	Input radius	-1.5	A	7.07
5	Display total area		R/S	22.93

Notes

THE GAME OF NIMB



Nimb is a game virtually every computer knows and loves. The HP–65 is no exception. The rules are quite simple. Starting with a total of fifteen objects (or in this case the number fifteen) each player alternately subtracts one, two, or three, until only one is left. The player forced to take the last one is the loser. A negative sign indicates that it is the users move while a positive display indicates it is the HP–65's turn.

As the challenger you are allowed to make the first move. It is possible to win but you must remember that your HP–65 is a master at this game and will not allow you to make an error and win.* Simply follow the user instruction form to play the game.

Notes: The algorithm used in this game is general. Any positive integer value may be used for a starting value. Simply store it in register 1 after pressing A.

*The HP–65 expects only chivalrous opponents. You could cheat by selecting numbers other than one, two or three.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program			
2	Start game		A	-15
3	User moves	1, 2, or 3	B	
4	HP–65 moves		C	
5	Go to 3 until game is over			
6	Turn machine around to see			
	message.			
7	For new case go to 2.			

EXAMPLE USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program (Nimb; as shown				
	on page 7 of this manual)				
2	Start game		A		-15
3	User takes 3	3	B		12
4	HP-65 moves		C		-9
3	User takes 2	2	B		7
4	HP-65 moves		C		-5
3	User takes 3	3	B		2
4	HP-65 moves		C		-1
3	User takes last 1	1	B		55178
	HP-65 has won				
6	Turn machine around for				
	message (BLISS).				

USER DIAGNOSTIC PROGRAM I

This program is designed to locate malfunctions in the operations of flags, relational operators, decrement and skip on zero, subroutine calls and tangent. In case calculator malfunctions are suspected, this program can be run as a check. In this way malfunctions can be specifically identified.

Examples:

To operate, turn the calculator OFF, then ON, enter the program as shown on page 7, and press **R/S** . You should see -8.888888888 -88. If not, the digit displayed corresponds to the calculator malfunction in the following listing e.g., a halt with 2 displayed means a stop after "f" "TF 2."

DISPLAY	CALCULATOR MALFUNCTION
0	g, $x \neq y$
1	f, TF 1 (with flag clear)
2	f, TF 2 (With flag clear)
3	g, $x \leqslant y$
4	f^{-1}, TF 1 (with flag set)
5	f^{-1}, TF 2 (with flag set)
6	g, $x = y$
7	f^{-1}, SF 1
8	f^{-1}, SF 2
9	g, $x > y$
-1	DSZ

Notes

USER DIAGNOSTIC PROGRAM II

This program is designed to locate malfunctions in

–, +, x, ÷, f 0–9, f^{-1} 0–9, f •, f^{-1} •, g •,

STO + 6, STO 8, RCL 8, RCL 6, CHS, EEX and R/S.

In case calculator malfunctions are suspected this program can be run as a check. In this way malfunctions can be specifically identified.

To operate, turn the calculator OFF, then ON, enter program as shown on page 7, and press **R/S**. You should see –8.888888888 –88. If not, turn the calculator OFF, then ON, reload the card, and step through the program **SST**, comparing displayed answers to the numbers in the display column of the following list. If answers do not match there is an error in the specified calculator function or operation.

DISPLAY	CALCULATOR FUNCTION CHECKED
0.00	
0.000000000 00	DSP 9
7.	
7.000000000 00	Lift Enable
1.945910149 00	f, LN
2.891227832 -01	f, LOG
5.377013885 -01	f, $\sqrt{x}$
9.384521785 -03	f, SIN
9.999999866 -01	f, COS
1.745506463 -02	f, TAN
1.000152325 00	f^{-1}, SIN
9.998476982 -01	g, $1/x$
9.999900006 -01	f^{-1}, COS
4.499971354 01	f^{-1}, TAN
2.222236368 -02	g, $1/x$
1.022471120 00	f^{-1}, LN, (e^x)
1.053103655 01	f^{-1}, LOG (10^x)
1.109027308 02	f^{-1}, $\sqrt{x}$ (x^2)
1.053103655 01	g, LST X

DISPLAY	CALCULATOR FUNCTION CHECKED
1.114016087 02	f, R→P
1.112406000 02	f,→D.MS
1.051581606 01	f,$^{-1}$ R→P
1.086615556 01	f,$^{-1}$→D.MS
1.224126000 02	f,D.MS+
2.	
–2.	CHS
–2.000000000 00	
2.000000000 00	g, ABS
1.498484464 04	g, y^x
3.141592654 00	g, π
4.769824191 03	÷
4.769000000 03	f, INT
1.124100000 04	f, →OCT
5.	
5.000000000 00	
1.200000000 02	g, n!
1.112100000 04	–
4.689000000 03	f^{-1}, →OCT
4.688000000 03	STO 8, RCL 8, DSZ
1.000000000 00	f^{-1}, D.MS+
4.688000000 03	g, LST X
9.376000000 03	STO + 6
9.	
–9.	CHS
–9.4	Program constant definition
–9.48	
–9.480	
–9.4804	
–9.48047	
–9.480470	
–9.4804702	
–9.48047023	
–9.480470230	
–9.480470230 –00	EEX CHS
–9.480470230 –09	Read exponent into
–9.480470230 –92	X register from memory
–8.888888888 –88	x
–8.888888888 –88	f^{-1}, INT
0.000000000 00	g, R↓
0.000000000 00	g, x⇄y
–8.888888888 –88	g, R↑
–8.888888888 –88	g, NOP
–8.888888888 –88	R/S

PROGRAM LISTINGS

DAY OF THE WEEK

CODE	KEYS
23	LBL
11	A
33 01	STO 1
24	RTN
23	LBL
12	B
33 02	STO 2
24	RTN
23	LBL
13	C
33 03	STO 3
24	RTN
23	LBL
14	D
34 01	RCL 1
35	g
04	$^1/x$
83	·
06	6
61	+
31	f
83	INT
33 06	STO 6
01	1
02	2
71	x
34 01	RCL 1
61	+
33 04	STO 4
34 03	RCL 3
34 06	RCL 6
51	–
33 05	STO 5
43	EEX
02	2

CODE	KEYS
81	÷
31	f
83	INT
35 00	g LST X
04	4
81	÷
31	f
83	INT
35 07	g x⇄y
51	–
34 05	RCL 5
05	5
71	x
04	4
81	÷
31	f
83	INT
61	+
34 04	RCL 4
01	1
61	+
01	1
03	3
71	x
05	5
81	÷
31	f
83	INT
61	+
34 02	RCL 2
61	+
01	1
51	–
41	↑
41	↑

CODE	KEYS
07	7
81	÷
31	f
83	INT
07	7
71	x
51	–
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	M	R_4	M′	R_7	
R_2	D	R_5	Y′	R_8	
R_3	Y	R_6	1 or 0	R_9	

MEAN, STANDARD DEVIATION, STANDARD ERROR

CODE	KEYS
00	0
33 01	STO 1
33 02	STO 2
33 03	STO 3
84	R/S
23	LBL
11	A
33	STO
61	+
02	2
32	f^{-1}
09	$\sqrt{x}$
33	STO
61	+
03	3
34 01	RCL 1
01	1
61	+
33 01	STO 1
24	RTN
23	LBL
12	B
34 02	RCL 2
34 01	RCL 1
81	÷
24	RTN
23	LBL
13	C
34 03	RCL 3
34 02	RCL 2
34 01	RCL 1
81	÷
32	f^{-1}
09	$\sqrt{x}$
34 01	RCL 1

CODE	KEYS
71	x
51	–
34 01	RCL 1
81	÷
31	f
09	$\sqrt{x}$
34 01	RCL 1
34 01	RCL 1
01	1
51	–
81	÷
31	f
09	$\sqrt{x}$
71	x
24	RTN
35 07	g x⇄y
84	R/S
23	LBL
14	D
13	C
34 01	RCL 1
31	f
09	$\sqrt{x}$
81	÷
35 07	g x⇄y
35 00	g LST X
81	÷
35 07	g x⇄y
84	R/S
35 07	g x⇄y
24	RTN
23	LBL
15	E
33	STO
51	–

CODE	KEYS
02	2
32	f^{-1}
09	$\sqrt{x}$
33	STO
51	–
03	3
34 01	RCL 1
01	1
51	–
33 01	STO 1
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	n	R_4	R_7
R_2	Σx_i	R_5	R_8
R_3	Σx_i^2	R_6	R_9

GREAT CIRCLE NAVIGATION

CODE	KEYS
32	f^{-1}
51	SF 1
31	f
43	REG
44	CLX
35	g
41	DEG
84	R/S
23	LBL
11	A
32	f^{-1}
03	→D.MS
34 01	RCL 1
33 02	STO 2
35 07	g x⇄y
33 01	STO 1
84	R/S
23	LBL
12	B
32	f^{-1}
03	→D.MS
34 03	RCL 3
33 04	STO 4
35 07	g x⇄y
33 03	STO 3
84	R/S
23	LBL
13	C
34 04	RCL 4
34 03	RCL 3
51	–
41	↑
31	f
04	SIN
00	0

CODE	KEYS
35 24	g x>y
31	f
51	SF 1
61	+
44	CLX
61	+
31	f
05	COS
34 02	RCL 2
31	f
05	COS
33 06	STO 6
71	x
34 01	RCL 1
31	f
05	COS
71	x
34 01	RCL 1
31	f
04	SIN
33 07	STO 7
34 02	RCL 2
31	f
04	SIN
33 08	STO 8
71	x
61	+
32	f^{-1}
05	COS
41	↑
41	↑
06	6
00	0
71	x
24	RTN

CODE	KEYS
23	LBL
14	D
03	3
06	6
00	0
13	C
35 08	g R↓
41	↑
31	f
05	COS
34 08	RCL 8
71	x
34 07	RCL 7
35 07	g x⇄y
51	–
35 07	g x⇄y
31	f
04	SIN
81	÷
34 06	RCL 6
81	÷
32	f^{-1}
05	COS
31	f
61	TF 1
51	–
35 01	g NOP
32	f^{-1}
51	SF 1
84	R/S

R_1	LAT_D	R_4	LNG_S	R_7	Used
R_2	LAT_S	R_5	0	R_8	Used
R_3	LNG_D	R_6	Used	R_9	Used

INTEGER BASE CONVERSION

CODE	KEYS
23	LBL
11	A
33 02	STO 2
24	RTN
23	LBL
12	B
33 03	STO 3
24	RTN
23	LBL
13	C
34 02	RCL 2
33 07	STO 7
35 08	g R↓
71	x
00	0
33 01	STO 1
01	1
33 05	STO 5
83	•
01	1
33 04	STO 4
33 06	STO 6
23	LBL
01	1
35 00	g LST X
34 04	RCL 4
71	x
34 07	RCL 7
33	STO
71	x
05	5
35 08	g R↓
35	g
06	ABS
01	1

CODE	KEYS
35 22	g x≤y
22	GTO
01	1
35 00	g LST X
23	LBL
02	2
34 04	RCL 4
81	÷
41	↑
31	f
83	INT
51	–
35 00	g LST X
34 05	RCL 5
34 07	RCL 7
81	÷
33 05	STO 5
71	x
33	STO
61	+
01	1
44	CLX
35 07	g x⇄y
35 21	g x≠y
22	GTO
02	2
33 07	STO 7
34 01	RCL 1
31	f
83	INT
23	LBL
03	3
41	↑
41	↑
34 03	RCL 3

CODE	KEYS
81	÷
31	f
83	INT
33 05	STO 5
34 03	RCL 3
71	x
51	–
34 06	RCL 6
34 04	RCL 4
81	÷
33 06	STO 6
71	x
33	STO
61	+
07	7
00	0
34 05	RCL 5
35 21	g x≠y
22	GTO
03	3
34 07	RCL 7
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	Used	R_4	Used	R_7	Used
R_2	B_2	R_5	Used	R_8	
R_3	B_1	R_6	Used	R_9	Used

BODY SURFACE AREA (BOYD)

CODE	KEYS
23	LBL
11	A
00	0
35 07	g x⇄y
35 24	g x>y
33 06	STO 6
24	RTN
42	CHS
02	2
83	•
05	5
04	4
71	x
33 06	STO 6
24	RTN
23	LBL
12	B
00	0
35 07	g x⇄y
35 24	g x>y
33 05	STO 5
24	RTN
02	2
83	•
02	2
42	CHS
81	÷
33 05	STO 5
24	RTN
23	LBL
13	C
83	•
07	7
02	2
08	8

CODE	KEYS
05	5
41	↑
83	•
00	0
01	1
08	8
08	8
34 05	RCL 5
43	EEX
03	3
71	x
41	↑
35 08	g R↓
31	f
08	LOG
71	x
51	–
35	g
05	y^x
34 06	RCL 6
83	•
03	3
35	g
05	y^x
71	x
03	3
83	•
02	2
00	0
07	7
71	x
43	EEX
04	4
81	÷
24	RTN

CODE	KEYS
23	LBL
14	D
43	EEX
02	2
71	x
33	STO
09	9
35 00	g LST X
81	÷
24	RTN
23	LBL
15	E
13	C
34	RCL
09	9
35 07	g x⇄y
81	÷
43	EEX
02	2
81	÷
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	R_4	R_7
R_2	R_5 Wt. (kg)	R_8
R_3	R_6 Ht. (cm)	R_9 100 CO(l/min)

PI NETWORK IMPEDANCE MATCHING

CODE	KEYS
31	f
42	STK
21	DSP
04	4
23	LBL
01	1
32	f^{-1}
51	SF 1
24	RTN
84	R/S
23	LBL
11	A
32	f^{-1}
61	TF 1
33 02	STO 2
84	R/S
33 01	STO 1
22	GTO
01	1
23	LBL
12	B
32	f^{-1}
61	TF 1
33 04	STO 4
84	R/S
33 03	STO 3
22	GTO
01	1
23	LBL
13	C
34 04	RCL 4
34 01	RCL 1
81	÷
31	f
61	TF 1

CODE	KEYS
22	GTO
00	0
34 02	RCL 2
34 01	RCL 1
81	÷
34 04	RCL 4
41	↑
71	x
01	1
61	+
33 05	STO 5
71	x
01	1
51	–
31	f
09	$\sqrt{x}$
34 02	RCL 2
81	÷
33 06	STO 6
22	GTO
00	0
23	LBL
14	D
13	C
34 02	RCL 2
34 06	RCL 6
71	x
34 04	RCL 4
81	÷
01	1
61	+
34 04	RCL 4
34 01	RCL 1
71	x
34 05	RCL 5

CODE	KEYS
81	÷
71	x
23	LBL
00	0
35	g
02	π
02	2
71	x
34 03	RCL 3
71	x
81	÷
22	GTO
01	1
23	LBL
15	E
31	f
51	SF 1
84	R/S
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	R_1	R_4	Q	R_7	
R_2	R_2	R_5	Used	R_8	
R_3	f	R_6	Used	R_9	

EDM SLOPE REDUCTION—GIVEN Δ ELEVATION

CODE	KEYS
23	LBL
11	A
33 01	STO 1
02	2
00	0
09	9
00	0
06	6
00	0
00	0
00	0
33 05	STO 5
34 01	RCL 1
24	RTN
23	LBL
12	B
33 01	STO 1
06	6
03	3
07	7
08	8
02	2
00	0
00	0
33 05	STO 5
34 01	RCL 1
24	RTN
23	LBL
13	C
33 02	STO 2
24	RTN
23	LBL
13	C
33 03	STO 3
24	RTN

CODE	KEYS
23	LBL
14	D
33	STO
61	+
02	2
33 07	STO 7
84	R/S
23	LBL
14	D
33	STO
61	+
03	3
84	R/S
23	LBL
14	D
33 04	STO 4
24	RTN
23	LBL
15	E
34 01	RCL 1
32	f^{-1}
09	$\sqrt{x}$
34 03	RCL 3
34 02	RCL 2
51	−
32	f^{-1}
09	$\sqrt{x}$
51	−
34 05	RCL 5
34 02	RCL 2
61	+
81	÷
34 05	RCL 5
34 03	RCL 3
61	+

CODE	KEYS
81	÷
31	f
09	$\sqrt{x}$
33 06	STO 6
34 05	RCL 5
34 07	RCL 7
61	+
71	x
84	R/S
34 06	RCL 6
34 05	RCL 5
71	x
84	R/S
34 06	RCL 6
34 04	RCL 4
34 05	RCL 5
61	+
71	x
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1 S Dist	R_4 E_S	R_7 E_1
R_2 HI DM + E_1	R_5 R	R_8
R_3 HT Rft + E_2	R_6 Used	R_9

TEMPERATURE CONVERSION

CODE	KEYS
32	f^{-1}
51	SF 1
84	R/S
23	LBL
12	B
41	↑
31	f
61	TF 1
22	GTO
00	0
05	5
71	x
09	9
81	÷
22	GTO
11	A
23	LBL
00	0
01	1
83	•
08	8
33	STO
71	x
01	1
22	GTO
11	A
23	LBL
13	C
41	↑
31	f
61	TF 1
22	GTO
02	2
03	3
02	2

CODE	KEYS
51	–
05	5
71	x
09	9
81	÷
22	GTO
14	D
23	LBL
02	2
01	1
83	•
08	8
33	STO
71	x
01	1
04	4
05	5
09	9
83	•
06	6
07	7
33	STO
51	–
01	1
22	GTO
11	A
23	LBL
14	D
41	↑
02	2
07	7
03	3
83	•
01	1
05	5

CODE	KEYS
31	f
61	TF 1
22	GTO
02	2
61	+
22	GTO
11	A
23	LBL
02	2
33	STO
51	–
01	1
23	LBL
11	A
32	f^{-1}
61	TF 1
22	GTO
06	6
34 01	RCL 1
32	f^{-1}
51	SF 1
24	RTN
23	LBL
06	6
33 01	STO 1
00	0
31	f
51	SF 1
24	RTN
35 01	g NOP

R_1 Temp K	R_4	R_7
R_2	R_5	R_8
R_3	R_6	R_9

WEIGHT-MASS CONVERSION

CODE	KEYS
23	LBL
11	A
00	0
35 23	g x=y
22	GTO
00	0
35 08	g R↓
01	1
06	6
71	x
12	B
23	LBL
00	0
01	1
06	6
33	STO
81	÷
04	4
35 08	g R↓
23	LBL
12	B
02	2
08	8
83	•
03	3
04	4
09	9
05	5
02	2
03	3
33 06	STO 6
71	x
00	0
35 21	g x≠y
35 08	g R↓

CODE	KEYS
14	D
34 06	RCL 6
33	STO
81	÷
04	4
35 08	g R↓
14	D
23	LBL
13	C
00	0
35 23	g x=y
34 04	RCL 4
84	R/S
35 08	g R↓
33 04	STO 4
00	0
41	↑
84	R/S
23	LBL
14	D
00	0
35 23	g x=y
22	GTO
02	2
35 08	g R↓
43	EEX
03	3
81	÷
13	C
23	LBL
02	2
43	EEX
03	3
33	STO
71	x

CODE	KEYS
04	4
35 08	g R↓
13	C
23	LBL
15	E
03	3
02	2
83	•
01	1
07	7
04	4
33 06	STO 6
71	x
00	0
35 21	g x≠y
35 08	g R↓
11	A
34 06	RCL 6
33	STO
81	÷
04	4
35 08	g R↓
11	A
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	R_4 Mass Kg	R_7
R_2	R_5	R_8
R_3	R_6 Used	R_9 Used

VOLUME CONVERSIONS

CODE	KEYS
23	LBL
11	A
03	3
83	•
07	7
08	8
05	5
04	4
33 06	STO 6
71	x
00	0
35 21	g x≠y
35 08	g R↓
13	C
34 06	RCL 6
33	STO
81	÷
05	5
35 08	g R↓
13	C
23	LBL
12	B
01	1
83	•
02	2
00	0
00	0
09	9
05	5
33 06	STO 6
71	x
00	0
35 21	g x≠y
35 08	g R↓
11	A

CODE	KEYS
34 06	RCL 6
33	STO
81	÷
05	5
35 08	g R↓
11	A
23	LBL
13	C
00	0
35 23	g x=y
34 05	RCL 5
84	R/S
35 08	g R↓
33 05	STO 5
00	0
41	↑
84	R/S
23	LBL
14	D
00	0
35 23	g x=y
22	GTO
02	2
35 08	g R↓
43	EEX
03	3
81	÷
13	C
23	LBL
02	2
43	EEX
03	3
33	STO
71	x
05	5

CODE	KEYS
35 08	g R↓
13	C
23	LBL
15	E
01	1
06	6
83	•
03	3
08	8
07	7
00	0
06	6
04	4
33 06	STO 6
71	x
00	0
35 21	g x≠y
35 08	g R↓
14	D
34 06	RCL 6
33	STO
81	÷
05	5
35 08	g R↓
14	D
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	R_4	R_7
R_2	R_5 Volume (liters)	R_8
R_3	R_6 Used	R_9 Used

COMPOUND INTEREST

CODE	KEYS
00	0
33 08	STO 8
84	R/S
23	LBL
11	A
35	g
83	DSZ
33 01	STO 1
24	RTN
34 04	RCL 4
34 03	RCL 3
81	÷
31	f
07	LN
34 02	RCL 2
31	f
07	LN
81	÷
33 01	STO 1
24	RTN
23	LBL
12	B
35	g
83	DSZ
22	GTO
02	2
34 04	RCL 4
34 03	RCL 3
81	÷
34 01	RCL 1
35	g
04	$^1/x$
35	g
05	y^x
33 02	STO 2

CODE	KEYS
01	1
51	–
43	EEX
02	2
71	x
24	RTN
23	LBL
02	2
41	↑
41	↑
43	EEX
02	2
81	÷
01	1
61	+
33 02	STO 2
35 07	g x⇄y
24	RTN
23	LBL
13	C
35	g
83	DSZ
33 03	STO 3
24	RTN
34 04	RCL 4
34 02	RCL 2
34 01	RCL 1
35	g
05	y^x
81	÷
33 03	STO 3
24	RTN
23	LBL
14	D
35	g

CODE	KEYS
83	DSZ
33 04	STO 4
24	RTN
34 03	RCL 3
34 02	RCL 2
34 01	RCL 1
35	g
05	y^x
71	x
33 04	STO 4
24	RTN
23	LBL
15	E
41	↑
01	1
33 08	STO 8
35 07	g x⇄y
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	n	R_4	FV	R_7	
R_2	1 + i/100	R_5		R_8	DSZ
R_3	PV	R_6		R_9	

LOAN REPAYMENT

CODE	KEYS
23	LBL
11	A
33 07	STO 7
71	x
33 01	STO 1
24	RTN
23	LBL
12	B
33 02	STO 2
43	EEX
02	2
33 05	STO 5
24	RTN
23	LBL
13	C
32	f^{-1}
71	SF 2
31	f
61	TF 1
22	GTO
01	1
33 03	STO 3
24	RTN
23	LBL
14	D
31	f
61	TF 1
22	GTO
01	1
33 04	STO 4
24	RTN
23	LBL
15	E
31	f
51	SF 1

CODE	KEYS
31	f
71	SF 2
24	RTN
23	LBL
01	1
32	f^{-1}
51	SF 1
34 02	RCL 2
34 07	RCL 7
81	÷
34 05	RCL 5
81	÷
01	1
61	+
34 01	RCL 1
35	g
05	y^x
33 06	STO 6
01	1
51	−
34 06	RCL 6
81	÷
34 07	RCL 7
71	x
34 02	RCL 2
81	÷
34 05	RCL 5
71	x
32	f^{-1}
81	TF 2
22	GTO
03	3
34 03	RCL 3
35 07	g x⇄y
81	÷

CODE	KEYS
22	GTO
04	4
23	LBL
03	3
34 04	RCL 4
71	x
23	LBL
04	4
33 08	STO 8
21	DSP
83	•
02	2
34 05	RCL 5
71	x
83	•
05	5
61	+
31	f
83	INT
34 05	RCL 5
81	÷
84	R/S
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	yn	R_4	PMT	R_7	n
R_2	i	R_5	100	R_8	PV or PMT
R_3	PV	R_6	$(1 + i/100)^{yxn}$	R_9	

RECONCILE CHECKING ACCOUNT

CODE	KEYS
23	LBL
11	A
35	g
83	DSZ
22	GTO
01	1
34 01	RCL 1
34 01	RCL 1
34 02	RCL 2
51	–
34 04	RCL 4
61	+
23	LBL
09	9
84	R/S
35 07	g x⇄y
22	GTO
09	9
23	LBL
01	1
33 01	STO 1
24	RTN
23	LBL
12	B
35	g
83	DSZ
22	GTO
02	2
34 03	RCL 3
34 02	RCL 2
22	GTO
09	9
23	LBL
02	2
41	↑

CODE	KEYS
34 03	RCL 3
01	1
61	+
33 03	STO 3
35 07	g x⇄y
33	STO
61	+
02	2
22	GTO
09	9
23	LBL
13	C
35	g
83	DSZ
22	GTO
03	3
34 05	RCL 5
34 04	RCL 4
22	GTO
09	9
23	LBL
03	3
41	↑
34 05	RCL 5
01	1
61	+
33 05	STO 5
35 07	g x⇄y
33	STO
61	+
04	4
22	GTO
09	9
23	LBL
14	D

CODE	KEYS
31	f
43	REG
31	f
42	STK
21	DSP
83	•
02	2
24	RTN
23	LBL
15	E
41	↑
01	1
33 08	STO 8
35 08	g R↓
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1 STATE. BAL.	R_4 Σ OUT DEPOSITS	R_7
R_2 Σ OUT CHECKS	R_5 # OUT DEPOSITS	R_8 DSZ
R_3 # OUT CHECKS	R_6	R_9

ITERATIVE SOLUTION OF f(x) = 0

CODE	KEYS
33 03	STO 3
84	R/S
23	LBL
12	B
33 01	STO 1
15	E
33 05	STO 5
84	R/S
23	LBL
13	C
33 02	STO 2
15	E
33 06	STO 6
84	R/S
23	LBL
14	D
34 02	RCL 2
34 02	RCL 2
34 01	RCL 1
51	–
34 06	RCL 6
34 05	RCL 5
51	–
81	÷
34 06	RCL 6
71	x
51	–
33 04	STO 4
51	–
35	g
06	ABS
34 03	RCL 3
35 24	g x>y
34 04	RCL 4
84	R/S

CODE	KEYS
34 04	RCL 4
15	E
34 06	RCL 6
35 07	g x⇄y
71	x
00	0
35 22	g x≤y
22	GTO
01	1
34 02	RCL 2
33 01	STO 1
34 06	RCL 6
33 05	STO 5
22	GTO
02	2
23	LBL
01	1
02	2
33	STO
81	÷
05	5
23	LBL
02	2
34 04	RCL 4
33 02	STO 2
35 00	g LST X
33 06	STO 6
22	GTO
14	D
23	LBL
15	E
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

CODE	KEYS
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	x_1	R_4	x_0	R_7	
R_2	x_2	R_5	$f(x_1)$	R_8	
R_3	δ	R_6	$f(x_2)$	R_9	Scratch

QUADRATIC EQUATION

CODE	KEYS
23	LBL
11	A
33 03	STO 3
35 08	g $R\downarrow$
42	CHS
33 02	STO 2
35 07	g $x \rightleftarrows y$
41	↑
33 01	STO 1
61	+
81	÷
41	↑
32	f^{-1}
09	$\sqrt{x}$
34 03	RCL 3
34 01	RCL 1
81	÷
33 03	STO 3
51	–
24	RTN
23	LBL
12	B
31	f
09	$\sqrt{x}$
35 07	g $x \rightleftarrows y$
35	g
06	ABS
61	+
34 02	RCL 2
34 01	RCL 1
71	x
41	↑
35	g
06	ABS
81	÷

CODE	KEYS
71	x
84	R/S
34 03	RCL 3
35 07	g $x \rightleftarrows y$
81	÷
24	RTN
23	LBL
13	C
42	CHS
31	f
09	$\sqrt{x}$
35 07	g $x \rightleftarrows y$
84	R/S
35 07	g $x \rightleftarrows y$
24	RTN
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

CODE	KEYS
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	a	R_4		R_7	
R_2	–b	R_5		R_8	
R_3	c, c/a	R_6		R_9	

AREAS AND SOLUTION OF RIGHT TRIANGLES

CODE	KEYS
31	f
43	REG
00	0
84	R/S
23	LBL
11	A
41	↑
35	g
06	ABS
23	LBL
12	B
71	x
35	g
02	π
23	LBL
13	C
22	GTO
06	6
23	LBL
14	D
00	0
35 24	g x>y
31	f
51	SF 1
35 08	g R↓
35	g
06	ABS
33 02	STO 2
35 08	g R↓
33 03	STO 3
35 08	g R↓
33 04	STO 4
61	+
61	+
61	+

CODE	KEYS
02	2
81	÷
33 05	STO 5
41	↑
41	↑
34 02	RCL 2
51	–
71	x
34 05	RCL 5
34 03	RCL 3
51	–
71	x
34 05	RCL 5
34 04	RCL 4
51	–
71	x
31	f
09	$\sqrt{x}$
31	f
61	TF 1
42	CHS
35 01	g NOP
32	f^{-1}
51	SF 1
01	1
23	LBL
06	6
71	x
33	STO
61	+
01	1
35	g
06	ABS
84	R/S
34 01	RCL 1

CODE	KEYS
24	RTN
23	LBL
15	E
33 02	STO 2
35 08	g R↓
33 03	STO 3
35 09	g R↑
71	x
00	0
35 24	g x>y
31	f
51	SF 1
34 03	RCL 3
41	↑
71	x
34 02	RCL 2
41	↑
71	x
31	f
61	TF 1
42	CHS
35 01	g NOP
61	+
35	g
06	ABS
31	f
09	$\sqrt{x}$
32	f^{-1}
51	SF 1
24	RTN

R_1	Σ Area	R_4	Used	R_7	
R_2	Used	R_5	Used	R_8	
R_3	Used	R_6		R_9	Used

NIMB

CODE	KEYS
23	LBL
11	A
21	DSP
83	•
00	0
01	1
05	5
33 01	STO 1
31	f
51	SF 1
15	E
23	LBL
13	C
31	f
51	SF 1
00	0
33 03	STO 3
23	LBL
02	2
34 03	RCL 3
03	3
35 23	g x=y
01	1
12	B
23	LBL
01	1
01	1
33	STO
61	+
03	3
34 01	RCL 1
01	1
51	–
34 03	RCL 3
51	–

CODE	KEYS
04	4
81	÷
31	f
83	INT
35 00	g LST X
51	–
00	0
35 21	g x≠y
22	GTO
02	2
34 03	RCL 3
23	LBL
12	B
33	STO
51	–
01	1
34 01	RCL 1
00	0
35 21	g x≠y
22	GTO
15	E
32	f^{-1}
61	TF 1
22	GTO
08	8
21	DSP
83	•
01	1
03	3
05	5
00	0
07	7
83	•
01	1
84	R/S

CODE	KEYS
23	LBL
08	8
05	5
05	5
01	1
07	7
08	8
84	R/S
23	LBL
15	E
34 01	RCL 1
31	f
61	TF 1
42	CHS
35 01	g NOP
32	f^{-1}
51	SF 1
84	R/S
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP
35 01	g NOP

R_1	Total	R_4		R_7	
R_2		R_5		R_8	
R_3	n	R_6		R_9	Used

USER DIAGNOSTIC PROGRAM I

CODE	KEYS
11	A
35 21	g x≠y
00	0
84	R/S
31	f
61	TF 1
01	1
84	R/S
31	f
81	TF 2
02	2
84	R/S
12	B
35 22	g x≤y
03	3
84	R/S
32	f^{-1}
61	TF 1
04	4
84	R/S
32	f^{-1}
81	TF 2
05	5
84	R/S
13	C
35 23	g x=y
06	6
84	R/S
31	f
61	TF 1
07	7
84	R/S
31	f
81	TF 2
08	8

CODE	KEYS
84	R/S
14	D
35 24	g x>y
09	9
84	R/S
15	E
05	5
42	CHS
83	•
00	0
09	9
02	2
09	9
05	5
08	8
01	1
07	7
08	8
43	EEX
42	CHS
08	8
06	6
31	f
06	TAN
21	DSP
09	9
21	DSP
09	9
84	R/S
23	LBL
11	A
05	5
33 08	STO 8
31	f
42	STK

CODE	KEYS
24	RTN
23	LBL
12	B
34 08	RCL 8
31	f
51	SF 1
31	f
71	SF 2
22	GTO
15	E
23	LBL
13	C
32	f^{-1}
51	SF 1
32	f^{-1}
71	SF 2
22	GTO
15	E
23	LBL
14	D
44	CLX
23	LBL
15	E
35	g
83	DSZ
35 01	g NOP
24	RTN
01	1
42	CHS
84	R/S

R_1	R_4	R_7
R_2	R_5	R_8 DSZ
R_3	R_6	R_9 Used

USER DIAGNOSTIC PROGRAM II

CODE	KEYS
31	f
43	REG
21	DSP
09	9
07	7
31	f
07	LN
31	f
08	LOG
31	f
09	$\sqrt{x}$
31	f
04	SIN
31	f
05	COS
31	f
06	TAN
32	f^{-1}
04	SIN
35	g
04	$1/x$
32	f^{-1}
05	COS
32	f^{-1}
06	TAN
35	g
04	$1/x$
32	f^{-1}
07	LN
32	f^{-1}
08	LOG
32	f^{-1}
09	$\sqrt{x}$
35 00	g LST X
31	f

CODE	KEYS
01	R→P
31	f
03	→D.MS
32	f^{-1}
01	R→P
32	f^{-1}
03	→D.MS
31	f
02	D.MS+
02	2
42	CHS
35	g
06	ABS
35	g
05	y^x
35	g
02	π
81	÷
31	f
83	INT
31	f
00	→OCT
05	5
35	g
03	n!
51	–
32	f^{-1}
00	→OCT
33 08	STO 8
35	g
83	DSZ
34 08	RCL 8
32	f^{-1}
02	D.MS+
35 00	g LST X

CODE	KEYS
33	STO
61	+
06	6
34 06	RCL 6
61	+
71	x
09	9
42	CHS
83	•
04	4
08	8
00	0
04	4
07	7
00	0
02	2
03	3
00	0
43	EEX
42	CHS
09	9
02	2
71	x
32	f^{-1}
83	INT
35 08	g R↓
35 07	g x⇄y
35 09	g R↑
35 01	g NOP
84	R/S

R_1	0	R_4	0	R_7	0
R_2	0	R_5	0	R_8	DSZ
R_3	0	R_6	Used	R_9	Used

HEWLETT hp PACKARD

Sales, service and support in 172 centers in 65 countries
19310-19320 Pruneridge Ave., Cupertino, CA 95014
Offices in principal cities throughout the United States

00065-90207 Rev. 6/74