

HEWLETT  PACKARD

HP-65

STANDARD PAC

JUNE, 1974

The program material contained herein is supplied without representation or warranty of any kind. Hewlett-Packard Company therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

PRINTED IN U.S.A.

© HEWLETT-PACKARD, 1974

CONTENTS

	Page
Introduction	3
Format of User Instructions	4
Entering a Program	7
1. Day of the Week	8
2. Mean, Standard Deviation, Standard Error	12
3. Great Circle Navigation	14
4. Integer Base Conversion	17
5. Body Surface Area (Boyd)	18
6. PI Network Impedance Matching	20
7. Electronic Distance Measuring Instrument Slope Reduction—Given Δ Elevation	23
8. Temperature Conversion	26
9. Weight-Mass Conversion	28
10. Volume Conversions	30
11. Compound Interest	32
12. Loan Repayment	34
13. Reconcile Checking Account	38
14. Iterative Solution of $f(x) = 0$	42
15. Quadratic Equation	46
16. Areas and Solution of Right Triangle	48
17. The Game of NIMB	52
18. User Diagnostic Program I	54
19. User Diagnostic Program II	56
Program Listings	59

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 occasionally 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 [T]. 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	$C_1 \dots C_n$	[B] []	$C_1 \dots C_n$
5	Repeat 5 for each outstanding deposit	$D_1 \dots D_m$	[C] []	$D_1 \dots D_m$
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:

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

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

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

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

$$D = \text{Day of the Month}$$

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

*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		<input type="text"/>	
2	Input a date:		<input type="text"/>	
	Month (January=1 and		<input type="text"/>	
	December=12)	M	<input type="text"/>	M
	and Day of the Month	D	<input type="text"/>	D
	and Year	Y	<input type="text"/>	Y
3	Calculate Day of the Week		<input type="text"/>	Day
4	For new date go to Step 2 and		<input type="text"/>	
	change any or all of the values		<input type="text"/>	

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;		<input type="text"/>	
	as shown on page 7 of this		<input type="text"/>	
	manual)		<input type="text"/>	
2	Input the Date		<input type="text"/>	
	Month	8	<input type="text"/>	8.00
	and Day of the Month	27	<input type="text"/>	27.00
	and Year	1939	<input type="text"/>	1939.00
3	Compute Day of the Week		<input type="text"/>	0.00

10 STD-01B

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		<input type="text"/>	
	Week; as shown on page 7 of		<input type="text"/>	
	this manual)		<input type="text"/>	
2	Input the Date		<input type="text"/>	
	Month	5	<input type="text"/>	5.00
	and Day of the Month	24	<input type="text"/>	24.00
	and Year	1876	<input type="text"/>	1876.00
3	Compute Day of the Week		<input type="text"/>	3.00

MEAN, STANDARD DEVIATION, STANDARD ERROR

MEAN, STANDARD DEVIATION, STANDARD ERROR			STD 02A	
$\Sigma+$	\bar{x}	s_x	$s_{\bar{x}}$	$\Sigma-$

Given a set of data points:

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

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:

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

Standard deviation

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

Standard error of the mean

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

Alternate method

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

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		<input type="button" value="ON"/>	
2	Initialize		<input type="button" value="RTN"/> <input type="button" value="R/S"/>	0.00
3	Repeat 3 for each x (Eliminate any x values entered in error—x error)	$x_1 \dots x_n$	<input type="button" value="A"/> <input type="button" value="ON"/>	1...n
4	Compute \bar{x}		<input type="button" value="B"/> <input type="button" value="ON"/>	\bar{x}
5	Compute s_x and		<input type="button" value="C"/> <input type="button" value="ON"/>	s_x
6	s_x'		<input type="button" value="R/S"/> <input type="button" value="ON"/>	s_x'
7	Compute $s_{\bar{x}}$ and		<input type="button" value="D"/> <input type="button" value="ON"/>	$s_{\bar{x}}$
8	$s_{\bar{x}}'$		<input type="button" value="R/S"/> <input type="button" value="ON"/>	$s_{\bar{x}}'$
9	For a new case, go to 2		<input type="button" value="ON"/> <input type="button" value="ON"/>	

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 keystone 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)		<input type="text"/> <input type="text"/>	
2	Initialize		<input type="text"/> RTN <input type="text"/> R/S	0.00
3	Input ages	62	<input type="text"/> A <input type="text"/>	1.00
		84	<input type="text"/> A <input type="text"/>	2.00
		47	<input type="text"/> A <input type="text"/>	3.00
		58	<input type="text"/> A <input type="text"/>	4.00
		68	<input type="text"/> A <input type="text"/>	5.00
		60	<input type="text"/> A <input type="text"/>	6.00
		62	<input type="text"/> A <input type="text"/>	7.00
		59	<input type="text"/> A <input type="text"/>	8.00
		71	<input type="text"/> A <input type="text"/>	9.00
		73	<input type="text"/> A <input type="text"/>	10.00
4	Compute \bar{x}		<input type="text"/> B <input type="text"/>	64.40
5	Compute s_x		<input type="text"/> C <input type="text"/>	10.10
6	Compute s_x^*		<input type="text"/> R/S <input type="text"/>	9.58
7	Compute s_z		<input type="text"/> D <input type="text"/>	3.19
8	Compute s_z^*		<input type="text"/> R/S <input type="text"/>	3.03

GREAT CIRCLE NAVIGATION

GREAT CIRCLE NAVIGATION		STD 03A	
LAT	LNG	calc dist	calc hdg

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 $^{\circ}$ minutes, i.e., 15.30 means $15^{\circ} 30'$).

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} \rightarrow DMS.)

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:

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

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

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		<input type="text"/> <input type="text"/>	
2	Initialize		RTN <input type="text"/> R/S <input type="text"/>	0.00
3	Input starting latitude	(deg . min)	A <input type="text"/> <input type="text"/>	(dec. deg.)
	and input starting longitude	(deg . min)	B <input type="text"/> <input type="text"/>	(dec. deg.)
4	Input destination latitude	(deg . min)	A <input type="text"/> <input type="text"/>	(dec. deg.)
	and input destination longitude	(deg . min)	B <input type="text"/> <input type="text"/>	(dec. deg.)
5	Calc. Great Circle Dist		C <input type="text"/> <input type="text"/>	(naut. miles)
	and/or calc. initial heading		D <input type="text"/> <input type="text"/>	(dec. deg.)
6	Go to step 4 to calculate next leg		<input type="text"/> <input type="text"/>	
7	To reinitialize go to step 2.		<input type="text"/> <input type="text"/>	

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: Dist₁ = 225.91 Nautical miles Hdg₁ = 212.66°
 Dist₂ = 522.04 Nautical miles Hdg₂ = 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)		<input type="text"/> <input type="text"/>	
2	Initialize		RTN <input type="text"/> R/S <input type="text"/>	0.00
3	Input LAT Chicago	41.50	A <input type="text"/> <input type="text"/>	41.83
	Input LNG Chicago	87.36	B <input type="text"/> <input type="text"/>	87.60
4	Input LAT St. Louis	38.38	A <input type="text"/> <input type="text"/>	38.63
	Input LNG St. Louis	90.12	B <input type="text"/> <input type="text"/>	90.20
5	Calc. Great Circle Distance		C <input type="text"/> <input type="text"/>	225.91
	Calc. initial heading		D <input type="text"/> <input type="text"/>	212.66
4	Input LAT New Orleans	29.56	A <input type="text"/> <input type="text"/>	29.93
	Input LNG New Orleans	90.04	B <input type="text"/> <input type="text"/>	90.07
5	Calc. Great Circle Distance		C <input type="text"/> <input type="text"/>	522.04
	Calc. initial heading		D <input type="text"/> <input type="text"/>	179.24

INTEGER BASE CONVERSION

INTEGER BASE CONVERSION			STD 04B
B ₁	B ₂	n	

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 $2 \leq B_i \leq 10$ ($i = 1, 2$).

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		<input type="text"/>	
2	Input the base B_1 of n	B_1	A <input type="text"/>	B_1
	and the desired base B_2	B_2	B <input type="text"/>	B_2
3	Input n in B_1 and convert to B_2	$n(B_1)$	C <input type="text"/>	$n(B_2)$
4	For new n go to step 3		<input type="text"/>	
	For new base go to step 2		<input type="text"/>	

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.)		<input type="text"/>	
2	Input B_1	2	A <input type="text"/>	2.00
	and B_2	8	B <input type="text"/>	8.00
3	Input n and convert	110	C <input type="text"/>	6.00

BODY SURFACE AREA (BOYD)

BODY SURFACE AREA (Boyd)				STD 05A
HEIGHT (cm, in)	WEIGHT (kg, lb)	BSA (m ²)	CO (l/min)	CI (l/min/m ²)

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:

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

$$CI = CO/BSA$$

$$W = 1000 W_t$$

$$BSA = \text{Body surface area (square meters)}$$

$$W = \text{Body weight (grams)}$$

$$W_t = \text{Body weight (kilograms)}$$

$$H = \text{Height (centimeters)}$$

$$CI = \text{Cardiac index (liters/min/m}^2\text{)}$$

$$CO = \text{Cardiac output (liters/min)}$$

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

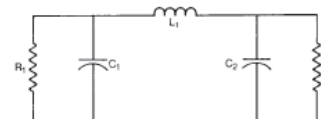
EXAMPLE USER INSTRUCTIONS

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:

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

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

$$X_{L1} = \frac{Q R_1}{Q^2 + 1} \left[1 + \frac{R_2}{Q X_{C2}} \right] \quad L_1 = \frac{X_{L1}}{2\pi f}$$

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		<input type="text"/> <input type="text"/>	
2	Initialize		RTN <input type="text"/> R/S <input type="text"/>	0.0000x10 ⁰
3	Input R ₁	R ₁ (ohms)	E <input type="text"/> A <input type="text"/>	R ₁ (ohms)
	and R ₂	R ₂ (ohms)	A <input type="text"/> <input type="text"/>	R ₂ (ohms)
	and f	f (Hz)	E <input type="text"/> B <input type="text"/>	f (Hz)
	and Q	Q	B <input type="text"/> <input type="text"/>	Q
4	Compute C ₁		E <input type="text"/> C <input type="text"/>	C ₁ (farads)
5	Compute C ₂		C <input type="text"/> <input type="text"/>	C ₂ (farads)
6	Compute L ₁		D <input type="text"/> <input type="text"/>	L ₁ (henrys)
7	Recall inputs (optional)		<input type="text"/> <input type="text"/>	
	R ₁		RCL <input type="text"/> 1 <input type="text"/>	R ₁ (ohms)
	and/or R ₂		RCL <input type="text"/> 2 <input type="text"/>	R ₂ (ohms)
	and/or f		RCL <input type="text"/> 3 <input type="text"/>	f (Hz)
	and/or Q		RCL <input type="text"/> 4 <input type="text"/>	Q
8	For new case change appropriate input in step 3.		<input type="text"/> <input type="text"/>	

Example:

$$\begin{aligned}
 R_1 &= 500 \quad R_2 = 50 \quad Q = 10 \quad 4 \times 10^6 \text{ (4MHz)} \\
 \text{Calculate } C_1 &= 7.9577 \times 10^{-10} \approx 796 \text{ pF} \\
 C_2 &= 2.4006 \times 10^{-9} \approx 2400 \text{ pF} \\
 L_1 &= 2.5639 \times 10^{-6} \approx 2.56 \text{ } \mu\text{H}
 \end{aligned}$$

EXAMPLE USER INSTRUCTIONS				
STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program		<input type="text"/> <input type="text"/>	
	(Pi Network Impedance Matching; as shown on page 7 of this manual)		<input type="text"/> <input type="text"/>	
2	Initialize		RTN <input type="text"/> R/S <input type="text"/>	0.0000x10 ⁰
3	Input R ₁	500	E <input type="text"/> A <input type="text"/>	5.0000x10 ²
	and R ₂	50	A <input type="text"/> <input type="text"/>	5.0000x10 ¹
	and f	4x10 ⁶	E <input type="text"/> B <input type="text"/>	4.0000x10 ⁶
	and Q	10	B <input type="text"/> <input type="text"/>	1.0000x10 ¹
4	Compute C ₁		E <input type="text"/> C <input type="text"/>	7.9577x10 ⁻¹⁰
5	Compute C ₂		C <input type="text"/> <input type="text"/>	2.4006x10 ⁻⁹
6	Compute L ₁		D <input type="text"/> <input type="text"/>	2.5639x10 ⁻⁶
7	Recall inputs (optional)		<input type="text"/> <input type="text"/>	
	R ₁		RCL <input type="text"/> 1 <input type="text"/>	5.0000x10 ²
	and/or R ₂		RCL <input type="text"/> 2 <input type="text"/>	5.0000x10 ¹
	and/or f		RCL <input type="text"/> 3 <input type="text"/>	4.0000x10 ⁶
	and/or Q		RCL <input type="text"/> 4 <input type="text"/>	1.0000x10 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.

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

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		<input type="button" value="A"/>	
2	Input slope distance in feet	S Dist(ft)	<input type="button" value="A"/>	S Dist(ft)
	or in meters	S Dist(m)	<input type="button" value="B"/>	S Dist(m)
3	Input height of distance meter	HI DM	<input type="button" value="C"/>	HI DM
4	Input height of reflector	HT Rft	<input type="button" value="C"/>	HT Rft
5	Input elevation at DM station	E_1	<input type="button" value="D"/>	E_1
6	Input elevation at Rft station	E_2	<input type="button" value="D"/>	E_2
7	Optional: Input specified elevation		<input type="button" value="D"/>	
		E_s	<input type="button" value="D"/>	E_s
8	Compute horizontal distance		<input type="button" value="E"/>	H Dist(E_1)
9	Optional: Compute H Dist at sea level		<input type="button" value="R/S"/>	H Dist(SL)
10	Optional: Compute H Dist at E_s		<input type="button" value="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)		<input type="text"/> <input type="text"/>	
2	Input slope distance in feet	10000	<input type="text"/> A <input type="text"/>	10000
3	Input height of distance meter	5.12	<input type="text"/> C <input type="text"/>	5.12
4	Input height of reflector	5.75	<input type="text"/> C <input type="text"/>	5.75
5	Input elevation at DM station	1000	<input type="text"/> D <input type="text"/>	1000
6	Input elevation at Rft station	3590.63	<input type="text"/> D <input type="text"/>	3590.63
7	Input specified elevation	2000	<input type="text"/> D <input type="text"/>	2000
8	Compute horizontal distance		<input type="text"/> E <input type="text"/>	9657.83
9	Compute H Dist at sea level		<input type="text"/> R/S <input type="text"/>	9657.37
10	Compute H Dist at specified elevation		<input type="text"/> R/S <input type="text"/>	9658.30

Note:

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

TEMPERATURE CONVERSION

TEMPERATURE CONVERSION				STD 08A
K	°R	°F	°C	

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

$$^{\circ}\text{C} = \frac{5}{9} (^{\circ}\text{F} - 32)$$

$$^{\circ}\text{F} = ^{\circ}\text{R} - 459.67$$

$$^{\circ}\text{R} = \frac{9}{5} \text{K}$$

$$\text{K} = ^{\circ}\text{C} + 273.15$$

GENERAL USER INSTRUCTIONS

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

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)		<input type="text"/> <input type="text"/>	
2	Initialize		<input type="text"/> <input type="text"/>	
3	Input °F	212°F	<input type="text"/> <input type="text"/>	0.00
4	Convert to K		<input type="text"/> <input type="text"/>	373.15 K

WEIGHT-MASS CONVERSION

WEIGHT-MASS CONVERSION				STD 09A
lbs	oz	kg	gm	slugs

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		<input type="text"/> <input type="text"/>	
2	Input either		<input type="text"/> <input type="text"/>	
	pounds	lbs	<input type="text"/> A <input type="text"/>	0.00
	or ounces	oz	<input type="text"/> B <input type="text"/>	0.00
	or kilograms	kg	<input type="text"/> C <input type="text"/>	0.00
	or grams	gm	<input type="text"/> D <input type="text"/>	0.00
	or slugs	slugs	<input type="text"/> E <input type="text"/>	0.00
3	Then convert to either		<input type="text"/> <input type="text"/>	
	pounds		<input type="text"/> A <input type="text"/>	lbs
	or ounces		<input type="text"/> B <input type="text"/>	oz
	or kilograms		<input type="text"/> C <input type="text"/>	kg
	or grams		<input type="text"/> D <input type="text"/>	gm
	or slugs		<input type="text"/> E <input type="text"/>	slugs
4	For new case go to 2.		<input type="text"/> <input type="text"/>	

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)		<input type="text"/> <input type="text"/>	
2	Input pounds	10	<input type="text"/> A <input type="text"/>	0.00
3	Convert to kilograms		<input type="text"/> C <input type="text"/>	4.54

VOLUME CONVERSIONS

VOLUME CONVERSIONS				STD 10A
US gal	IMP gal	liters	cc	cu in

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		<input type="text"/> <input type="text"/>	
2	Input either		<input type="text"/> <input type="text"/>	
	U.S. gallons	U.S. gal.	<input type="text"/> A <input type="text"/>	0.00
	or Imperial gallons	Imp. gal.	<input type="text"/> B <input type="text"/>	0.00
	or liters	l	<input type="text"/> C <input type="text"/>	0.00
	or cubic centimeters	cc	<input type="text"/> D <input type="text"/>	0.00
	or cubic inches	in ³	<input type="text"/> E <input type="text"/>	0.00
3	Then convert to either		<input type="text"/> <input type="text"/>	
	U.S. gallons		<input type="text"/> A <input type="text"/>	U.S. gal.
	or Imperial gallons		<input type="text"/> B <input type="text"/>	Imp. gal.
	or liters		<input type="text"/> C <input type="text"/>	l
	or cubic centimeters		<input type="text"/> D <input type="text"/>	cc
	or cubic inches		<input type="text"/> E <input type="text"/>	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)		<input type="text"/> <input type="text"/>	
2	Input cubic centimeters	2400	<input type="text"/> D <input type="text"/>	0.00
3	Convert to cubic inches		<input type="text"/> E <input type="text"/>	146.46

COMPOUND INTEREST

COMPOUND INTEREST				STD 11A	
n	i	PV	FV	COMPUTE	

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

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

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

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

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

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		<input type="text"/>	
2	Initialize		RTN <input type="text"/> R/S <input type="text"/>	
3	Input 3 of the following		<input type="text"/>	
	n	n	A <input type="text"/>	n
	and/or i	i(%)	B <input type="text"/>	i(%)
	and/or PV	PV	C <input type="text"/>	PV
	and/or FV	FV	D <input type="text"/>	FV
4	Compute the remaining variable		<input type="text"/>	
	n		E <input type="text"/> A <input type="text"/>	n
	or i		E <input type="text"/> B <input type="text"/>	i(%)
	or PV		E <input type="text"/> C <input type="text"/>	PV
	or FV		E <input type="text"/> D <input type="text"/>	FV
5	To modify problem go to 3		<input type="text"/>	
	and change appropriate input(s).		<input type="text"/>	

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)		<input type="text"/>	
2	Initialize		RTN <input type="text"/> R/S <input type="text"/>	0.00
3	Input		<input type="text"/>	
	n (n = 20 x 4)	80	A <input type="text"/>	80.00
	and i (i = 7 ÷ 4)	1.75	B <input type="text"/>	1.75
	and FV	15000	D <input type="text"/>	15000.00
4	Compute PV		E <input type="text"/> C <input type="text"/>	3744.02

LOAN REPAYMENT

LOAN REPAYMENT		STD 12A	
YRS per/yr	I%	PV	PMT CALC(P)

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		<input type="text"/>	
2	Input number of years	Y	<input type="text"/> 1 <input type="text"/>	Y
3	Input payment periods/year	n	<input type="text"/> A <input type="text"/>	Yn
4	Input annual interest	i(%)	<input type="text"/> B <input type="text"/>	100
5	Input either principal value	PV	<input type="text"/> C <input type="text"/>	PV
	or payment	PMT	<input type="text"/> D <input type="text"/>	PMT
6	Calculate either		<input type="text"/>	
	payment		<input type="text"/> E <input type="text"/> D	PMT
	or principal value		<input type="text"/> E <input type="text"/> C	PV
7	Perform steps 2 and 3, or step 4		<input type="text"/>	
	or step 5 for new values.		<input type="text"/>	
	Note: Unrounded values for		<input type="text"/>	
	PMT or PV calculations are		<input type="text"/>	
	stored in register 8 after		<input type="text"/>	
	calculation.		<input type="text"/>	

36 STD-12A

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		<input type="text"/>	
	; as shown on page 7 of this		<input type="text"/>	
	manual)		<input type="text"/>	
2	Input number of years	30	<input type="text"/> 1 <input type="text"/>	30.00
3	Input payment periods/year	4	<input type="text"/> A <input type="text"/>	120.00
4	Input annual interest	8.75	<input type="text"/> B <input type="text"/>	100.00
5	Input principal value	37500	<input type="text"/> C <input type="text"/>	37500.00
6	Calculate payment		<input type="text"/> E <input type="text"/> D	886.36
5	Input payment		<input type="text"/> D <input type="text"/>	886.36
6	Calculate principal value		<input type="text"/> E <input type="text"/> C	37499.86
	(the difference is due to		<input type="text"/>	
	accumulated % cent roundoff)		<input type="text"/>	
4	Input new interest	9.25	<input type="text"/> B <input type="text"/>	100.00
6	Calculate payment		<input type="text"/> E <input type="text"/> D	926.83
2	Input number of years	30	<input type="text"/> 1 <input type="text"/>	
3	Input payment periods/year	12	<input type="text"/> A <input type="text"/>	360.00
6	Calculate payment		<input type="text"/> E <input type="text"/> D	308.50

RECONCILE CHECKING ACCOUNT

RECONCILE CHECKING ACCOUNT				STD 13A	
F BAL	SUM	SUM	CLEAR	COMPUTE	
STATE BAL	OUT	CHK	OUT	DEP	

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		<input type="text"/>	
2	Initialize		<input type="text"/> D <input type="text"/>	0.00
3	Input statement balance	SB	<input type="text"/> A <input type="text"/>	SB
4	Repeat 4 for out checks	$C_1 \dots C_n$	<input type="text"/> B <input type="text"/>	$C_1 \dots C_n$
5	Repeat 5 for each outstanding deposit	$D_1 \dots D_m$	<input type="text"/> C <input type="text"/>	$D_1 \dots D_m$
6	Compute final balance		<input type="text"/> E <input type="text"/> A	FB
7	Recall statement balance		<input type="text"/> R/S <input type="text"/>	SB
	and/or sum of out checks		<input type="text"/> E <input type="text"/> B	ΣC_i
	and number of checks		<input type="text"/> R/S <input type="text"/>	n_c
	and/or sum of out deposit		<input type="text"/> E <input type="text"/> C	ΣD_i
	and number of deposits		<input type="text"/> R/S <input type="text"/>	m_D
8	To add additional checks go to 4, to add additional deposits go to 5.		<input type="text"/>	
9	For new case go to 2.		<input type="text"/>	

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)		<input type="text"/>	
2	Initialize		<input type="text"/> D <input type="text"/>	0.00
3	Input statement balance	432.96	<input type="text"/> A <input type="text"/>	432.96
4	Input outstanding checks		<input type="text"/>	
		47.82	<input type="text"/> B <input type="text"/>	47.82
		5.63	<input type="text"/> B <input type="text"/>	5.63
		25.00	<input type="text"/> B <input type="text"/>	25.00
		36.47	<input type="text"/> B <input type="text"/>	36.47
		96.02	<input type="text"/> B <input type="text"/>	96.02
5	Input outstanding deposits		<input type="text"/>	
		100.00	<input type="text"/> C <input type="text"/>	100.00
		256.03	<input type="text"/> C <input type="text"/>	256.03
6	Compute final balance		<input type="text"/> E <input type="text"/> A	578.05
7	Recall statement balance		<input type="text"/> R/S <input type="text"/>	432.96
	and/or sum of out checks		<input type="text"/> E <input type="text"/> B	210.94
	and number of out checks		<input type="text"/> R/S <input type="text"/>	5.00
	and/or sum of out deposits		<input type="text"/> E <input type="text"/> C	356.03
	and number of out deposits		<input type="text"/> R/S <input type="text"/>	2.00

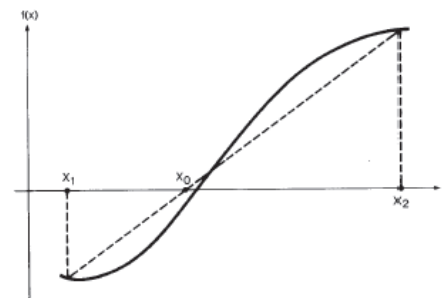
ITERATIVE SOLUTION OF $f(x)=0$

ITERATIVE SOLUTION OF $f(x)=0$					STD 14B
δ	x_1	x_2	SOLN x_0	$f(x)$	

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

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

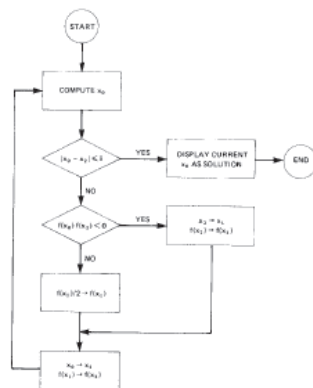


The estimates to be used for the next iteration are chosen according to the following rules such that the function values used at each iteration will always have opposite signs:

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 $(x_1, \frac{f(x_1)}{2})$.
2. Replace $(x_2, f(x_2))$ by $(x_0, f(x_0))$.

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

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:

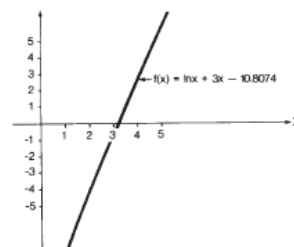
M. Dowell and P. Jarratt, "A Modified Regula Falsi Method for Computing the Root of an Equation," BIT 11, 1971, pp. 168-174.

GENERAL USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Enter program		GTO E	
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 If $f(x_2)$ has the same sign as $f(x_1)$, choose new x_2 and repeat 8.	x_2	C	$f(x_2)$
9	Compute solution x_0		D	x_0
10	Compute $f(x_0)$		E	$f(x_0)$
11	(Optional) input a new δ , then go to 9. (For a new $f(x)$, go to 2.) * x_2 may be chosen smaller than x_1 if necessary.	δ	A	δ

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 R₇ before we key in the function f(x).

$$\text{Answer: } x_0 = 3.21$$

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

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)		<input type="text"/> <input type="text"/>	
2	Store constant 10.8074 in R ₇	10.8074	<input type="text"/> STO <input type="text"/> 7	10.81
	Go to label E		<input type="text"/> GTO <input type="text"/> E	10.81
3	Switch to W/PRGM mode		<input type="text"/> <input type="text"/>	15
4	Key in steps defining f(x) = ln x + 3x - 10.8074		<input type="text"/> f <input type="text"/>	31
			<input type="text"/> LN <input type="text"/>	07
			<input type="text"/> g <input type="text"/> (1ST X)	35.00
			<input type="text"/> 3 <input type="text"/>	03
			<input type="text"/> x <input type="text"/>	71
			<input type="text"/> + <input type="text"/>	61
			<input type="text"/> RCL <input type="text"/> 7	34.07
			<input type="text"/> - <input type="text"/>	51
			<input type="text"/> RTN <input type="text"/>	24
5	Switch to RUN mode		<input type="text"/> <input type="text"/>	10.81
6	Input δ (say $\delta = 10^{-8}$)	10^{-8}	<input type="text"/> A <input type="text"/>	0.00
7	Input $x_1 = 1$	1	<input type="text"/> B <input type="text"/>	-7.81
8	Guess $x_2 = 2$	2	<input type="text"/> C <input type="text"/>	-4.11
	f(x ₂) has same sign as f(x ₁), guess $x_2 = 4$	4	<input type="text"/> C <input type="text"/>	2.58
9	Compute solution x_0 (takes about 12 seconds)		<input type="text"/> D <input type="text"/>	3.21
10	Compute f(x ₀)		<input type="text"/> E <input type="text"/>	-0.00
	Change display format to see it		<input type="text"/> DSP <input type="text"/> 9	-4×10^{-8}

QUADRATIC EQUATION

QUADRATIC EQUATION			STD 15B
a, b, c	D ≥ 0	D < 0	<input type="text"/>

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

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

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

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

$$\text{then the smaller root by } x_2 = \frac{c}{x_1 a}$$

$$\text{If } D = (b^2 - 4ac)/4a^2$$

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		<input type="text"/> <input type="text"/>	
2	Input coefficients		<input type="text"/> <input type="text"/>	
	a	a	<input type="text"/> f <input type="text"/>	a
	and b	b	<input type="text"/> f <input type="text"/>	b
	and c	c	<input type="text"/> A <input type="text"/>	D
3	If D ≥ 0 roots are real		<input type="text"/> B <input type="text"/>	root 1
			<input type="text"/> R/S <input type="text"/>	root 2
4	If D < 0 roots are complex		<input type="text"/> C <input type="text"/>	u
			<input type="text"/> R/S <input type="text"/>	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)		<input type="text"/> <input type="text"/>	
2	Input coefficients		<input type="text"/> <input type="text"/>	
	a	2	<input type="text"/> <input type="text"/>	2.00
	and b	5	<input type="text"/> <input type="text"/>	5.00
	and c	3	<input type="text"/> <input type="text"/>	0.06
3	Compute root 1		<input type="text"/> <input type="text"/>	-1.50
	Compute root 2		<input type="text"/> <input type="text"/>	-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)		<input type="text"/> <input type="text"/>	
2	Input coefficients		<input type="text"/> <input type="text"/>	
	a	2	<input type="text"/> <input type="text"/>	2.00
	and b	3	<input type="text"/> <input type="text"/>	3.00
	and c	4	<input type="text"/> <input type="text"/>	-1.44
4	Compute u		<input type="text"/> <input type="text"/>	-0.75
	Compute v		<input type="text"/> <input type="text"/>	1.20

AREAS AND SOLUTION OF RIGHT TRIANGLE

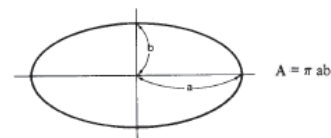


This program finds:

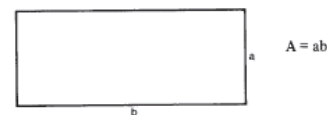
1. The area of a circle given the radius a



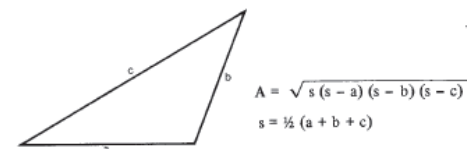
2. The area of an ellipse given a and b



3. The area of a rectangle given a and b



4. The area of a triangle given a, b, and 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		<input type="text"/>	
2	Initialize		<input type="text"/> RTN <input type="text"/> R/S	0.00
3	To find the area of a circle	\pm *a	<input type="text"/> A <input type="text"/>	Area
	or an ellipse	a	<input type="text"/> f <input type="text"/>	
		\pm *b	<input type="text"/> B <input type="text"/>	Area
	or a rectangle	a	<input type="text"/> f <input type="text"/>	
		\pm *b	<input type="text"/> C <input type="text"/>	Area
	or a triangle	a	<input type="text"/> f <input type="text"/>	
		b	<input type="text"/> f <input type="text"/>	
		\pm *c	<input type="text"/> D <input type="text"/>	Area
	or to find the third side of a right triangle	a	<input type="text"/> f <input type="text"/>	
		b or -c	<input type="text"/> E <input type="text"/>	side 3
4	For next area go to 3.		<input type="text"/>	
5	For total of areas previously computed (this step must directly follow an area computation)		<input type="text"/> R/S <input type="text"/>	Σ Areas
6	To start new sum go to 2.		<input type="text"/>	
	*Note: Input as a negative number if the area is to be subtracted from the total being accumulated.		<input type="text"/>	

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 (Area and Solution of Right Triangles; as shown on page 7 of this manual)		<input type="text"/>	
2	Initialize		<input type="text"/> RTN <input type="text"/> R/S	0.00
3-4	Compute third side of triangle		<input type="text"/>	
	Input side	3	<input type="text"/> f <input type="text"/>	3.00
	Input side	4	<input type="text"/> E <input type="text"/>	5.00
	Compute area of triangle		<input type="text"/>	
	Input side	3	<input type="text"/> f <input type="text"/>	3.00
	Input side	4	<input type="text"/> f <input type="text"/>	4.00
	Input side	5	<input type="text"/> D <input type="text"/>	6.00
	Compute area of rectangle		<input type="text"/>	
	Input side	6	<input type="text"/> f <input type="text"/>	6.00
	Input side	4	<input type="text"/> C <input type="text"/>	24.00
	Delete area of circle		<input type="text"/>	
	Input radius	-1.5	<input type="text"/> A <input type="text"/>	7.07
5	Display total area		<input type="text"/> R/S <input type="text"/>	22.93

THE GAME OF NIMB

NIMB			STD 17A
START	IF - YOU	IF + ME	<input type="text"/>

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		<input type="text"/>	
2	Start game		A <input type="text"/>	-15
3	User moves	1, 2, or 3	B <input type="text"/>	
4	HP-65 moves		C <input type="text"/>	
5	Go to 3 until game is over		<input type="text"/>	
6	Turn machine around to see message.		<input type="text"/>	
7	For new case go to 2.		<input type="text"/>	

EXAMPLE USER INSTRUCTIONS

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

USER DIAGNOSTIC PROGRAM I

USER DIAGNOSTIC PROGRAM I STD 18A

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

0
1
2
3
4
5
6
7
8
9
-1

CALCULATOR MALFUNCTION

g, $x \neq y$
f, TF 1 (with flag clear)
f, TF 2 (With flag clear)
g, $x \leq y$
f⁻¹, TF 1 (with flag set)
f⁻¹, TF 2 (with flag set)
g, $x = y$
f⁻¹, SF 1
f⁻¹, SF 2
g, $x > y$
DSZ

USER DIAGNOSTIC PROGRAM II

USER DIAGNOSTIC PROGRAM II STD 19A



This program is designed to locate malfunctions in

$-, +, \times, \div, f, 0-9, f^{-1}, 0-9, f^{-1}, g, +,$

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 -8.88888888 -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	$g, \frac{1}{x}$
9.999900006 -01	f^{-1} , COS
4.499971354 01	f^{-1} , TAN
2.222236368 -02	$g, \frac{1}{x}$
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	$f_i R \rightarrow P$
1.112406000 02	$f_i \rightarrow D.MS$
1.051581606 01	$f_i^{-1} R \rightarrow P$
1.086615556 01	$f_i^{-1} \rightarrow D.MS$
1.224126000 02	$f_i D.MS+$
2.	
-2.	CHS
-2.000000000 00	
2.000000000 00	$g_i ABS$
1.498484464 04	$g_i y^x$
3.141592654 00	$g_i \pi$
4.769824191 03	\div
4.769000000 03	$f_i INT$
1.124100000 04	$f_i \rightarrow OCT$
5.	
5.000000000 00	
1.200000000 02	$g_i n!$
1.112100000 04	$-$
4.689000000 03	$f_i^{-1}, \rightarrow OCT$
4.688000000 03	STO 8, RCL 8, DSZ
1.000000000 00	$f_i^{-1}, D.MS+$
4.688000000 03	$g_i LST X$
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 X register from memory
-9.480470230 -92	x
-8.888888888 -88	f_i^{-1}, INT
-8.888888888 -88	$g_i R\downarrow$
0.000000000 00	$g_i x \rightleftharpoons y$
0.000000000 00	$g_i R\uparrow$
-8.888888888 -88	$g_i NOP$
-8.888888888 -88	R/S
-8.888888888 -88	

PROGRAM LISTINGS

	Page
1. Day of the Week	60
2. Mean, Standard Deviation, Standard Error	61
3. Great Circle Navigation	62
4. Integer Base Conversion	63
5. Body Surface Area (Boyd)	64
6. PI Network Impedance Matching	65
7. EDM Slope Reduction—Given Δ Elevation	66
8. Temperature Conversion	67
9. Weight-Mass Conversion	68
10. Volume Conversions	69
11. Compound Interest	70
12. Loan Repayment	71
13. Reconcile Checking Account	72
14. Iterative Solution of $f(x) = 0$	73
15. Quadratic Equation	74
16. Areas and Solution of Right Triangle	75
17. NIMB	76
18. User Diagnostic Program I	77
19. User Diagnostic Program II	78

DAY OF THE WEEK

CODE	KEYS	CODE	KEYS	CODE	KEYS
23	LBL	81	\div	07	7
11	A	31	f	81	\div
33 01	STO 1	83	INT	31	f
24	RTN	35 00	g LST X	83	INT
23	LBL	04	4	07	7
12	B	81	\div	71	x
33 02	STO 2	31	f	51	—
24	RTN	83	INT	24	RTN
23	LBL	35 07	g $x \div y$	35 01	g NOP
13	C	51	—	35 01	g NOP
33 03	STO 3	34 05	RCL 5	35 01	g NOP
24	RTN	05	5	35 01	g NOP
23	LBL	71	x	35 01	g NOP
14	D	04	4	35 01	g NOP
34 01	RCL 1	81	\div	35 01	g NOP
35	g	31	f	35 01	g NOP
04	$1/x$	83	INT	35 01	g NOP
83	.	61	+	35 01	g NOP
06	6	34 04	RCL 4	35 01	g NOP
61	+	01	1	35 01	g NOP
31	f	61	+	35 01	g NOP
83	INT	01	1	35 01	g NOP
33 06	STO 6	03	3	35 01	g NOP
01	1	71	x	35 01	g NOP
02	2	05	5	35 01	g NOP
71	x	81	\div	35 01	g NOP
34 01	RCL 1	31	f	35 01	g NOP
61	+	83	INT	35 01	g NOP
33 04	STO 4	61	+	35 01	g NOP
34 03	RCL 3	34 02	RCL 2	35 01	g NOP
34 06	RCL 6	61	+		
51	—	01	1		
33 05	STO 5	51	—		
43	EEX	41	\uparrow		
02	2	41	\uparrow		

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	CODE	KEYS	CODE	KEYS
00	0	71	x	02	2
33 01	STO 1	51	—	32	f^{-1}
33 02	STO 2	34 01	RCL 1	09	\sqrt{x}
33 03	STO 3	81	\div	33	STO
84	R/S	31	f	51	—
23	LBL	09	\sqrt{x}	03	3
11	A	34 01	RCL 1	34 01	RCL 1
33	STO	34 01	RCL 1	01	1
61	+	01	1	51	—
02	2	51	—	33 01	STO 1
32	f^{-1}	81	\div	24	RTN
09	\sqrt{x}	31	f	35 01	g NOP
33	STO	09	\sqrt{x}	35 01	g NOP
61	+	71	x	35 01	g NOP
03	3	24	RTN	35 01	g NOP
34 01	RCL 1	35 07	g $x \div y$	35 01	g NOP
01	1	84	R/S	35 01	g NOP
61	+	23	LBL	35 01	g NOP
33 01	STO 1	14	D	35 01	g NOP
24	RTN	13	C	35 01	g NOP
23	LBL	34 01	RCL 1	35 01	g NOP
12	B	31	f	35 01	g NOP
34 02	RCL 2	09	\sqrt{x}	35 01	g NOP
34 01	RCL 1	81	\div	35 01	g NOP
81	\div	35 07	g $x \div y$	35 01	g NOP
24	RTN	35 00	g LST X	35 01	g NOP
23	LBL	81	\div	35 01	g NOP
13	C	35 07	g $x \div y$	35 01	g NOP
34 03	RCL 3	84	R/S	35 01	g NOP
34 02	RCL 2	35 07	g $x \div y$	35 01	g NOP
34 01	RCL 1	24	RTN	35 01	g NOP
81	\div	23	LBL		
32	f^{-1}	15	E		
09	\sqrt{x}	33	STO		
34 01	RCL 1	51	—		

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	CODE	KEYS	CODE	KEYS
32	f^{-1}	35 24	g $x > y$	23	LBL
51	SF 1	31	f	14	D
31	f	51	SF 1	03	3
43	REG	61	+	06	6
44	CLX	44	CLX	00	0
35	g	61	+	13	C
41	DEG	31	f	35 08	g R ↓
84	R/S	05	COS	41	↑
23	LBL	34 02	RCL 2	31	f
11	A	31	f	05	COS
32	f^{-1}	05	COS	34 08	RCL 8
03	→D.MS	33 06	STO 6	71	x
34 01	RCL 1	71	x	34 07	RCL 7
33 02	STO 2	34 01	RCL 1	35 07	g $x \div y$
35 07	g $x \div y$	31	f	51	—
33 01	STO 1	05	COS	35 07	g $x \div y$
84	R/S	71	x	31	f
23	LBL	34 01	RCL 1	04	SIN
12	B	31	f	81	\div
32	f^{-1}	04	SIN	34 06	RCL 6
03	→D.MS	33 07	STO 7	81	\div
34 03	RCL 3	34 02	RCL 2	32	f^{-1}
33 04	STO 4	31	f	05	COS
35 07	g $x \div y$	04	SIN	31	f
33 03	STO 3	33 08	STO 8	61	TF 1
84	R/S	71	x	51	—
23	LBL	61	+	35 01	g NOP
13	C	32	f^{-1}	32	f^{-1}
34 04	RCL 4	05	COS	51	SF 1
34 03	RCL 3	41	↑	84	R/S
51	—	41	↑		
41	↑	06	6		
31	f	00	0		
04	SIN	71	x		
00	0	24	RTN		

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	CODE	KEYS	CODE	KEYS
23	LBL	35 22	$g x \leq y$	81	\div
11	A	22	GTO	31	f
33 02	STO 2	01	1	83	INT
24	RTN	35 00	$g LST X$	33 05	STO 5
23	LBL	23	LBL	34 03	RCL 3
12	B	02	2	71	x
33 03	STO 3	34 04	RCL 4	51	-
24	RTN	81	\div	34 06	RCL 6
23	LBL	41	\uparrow	34 04	RCL 4
13	C	31	f	81	\div
34 02	RCL 2	83	INT	33 06	STO 6
33 07	STO 7	51	-	71	x
35 08	$g R \downarrow$	35 00	$g LST X$	33	STO
71	x	34 05	RCL 5	61	+
00	0	34 07	RCL 7	07	7
33 01	STO 1	81	\div	00	0
01	1	33 05	STO 5	34 05	RCL 5
33 05	STO 5	71	x	35 21	$g x \neq y$
83	.	33	STO	22	GTO
01	1	61	+	03	3
33 04	STO 4	01	1	34 07	RCL 7
33 06	STO 6	44	CLX	24	RTN
23	LBL	35 07	$g x \neq y$	35 01	g NOP
01	1	35 21	$g x \neq y$	35 01	g NOP
35 00	$g LST X$	22	GTO	35 01	g NOP
34 04	RCL 4	02	2	35 01	g NOP
71	x	33 07	STO 7	35 01	g NOP
34 07	RCL 7	34 01	RCL 1	35 01	g NOP
33	STO	31	f	35 01	g NOP
71	x	83	INT	35 01	g NOP
05	5	23	LBL		
35 08	$g R \downarrow$	03	3		
35	g	41	\uparrow		
06	ABS	41	\uparrow		
01	1	34 03	RCL 3		

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	CODE	KEYS	CODE	KEYS
23	LBL	05	5	23	LBL
11	A	41	\uparrow	14	D
00	0	83	.	43	EEX
35 07	$g x \neq y$	00	0	02	2
35 24	$g x > y$	01	1	71	x
33 06	STO 6	08	8	33	STO
24	RTN	08	8	09	9
42	CHS	34 05	RCL 5	35 00	$g LST X$
02	2	43	EEX	81	\div
83	.	03	3	24	RTN
05	5	71	x	23	LBL
41	\uparrow	41	\uparrow	15	E
71	x	35 08	$g R \downarrow$	13	C
33 06	STO 6	31	f	34	RCL
24	RTN	08	LOG	09	9
23	LBL	71	x	35 07	$g x \neq y$
12	B	51	-	81	\div
00	0	35	g	43	EEX
35 07	$g x \neq y$	05	y^x	02	2
35 24	$g x > y$	34 06	RCL 6	81	\div
33 05	STO 5	83	.	24	RTN
24	RTN	03	3	35 01	g NOP
02	2	35	g	35 01	g NOP
83	.	05	y^x	35 01	g NOP
02	2	71	x	35 01	g NOP
42	CHS	03	3	35 01	g NOP
81	\div	83	.	35 01	g NOP
33 05	STO 5	02	2	35 01	g NOP
24	RTN	00	0	35 01	g NOP
23	LBL	07	7	35 01	g NOP
13	C	71	x		
83	.	43	EEX		
07	7	04	4		
02	2	81	\div		
08	8	24	RTN		

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	CODE	KEYS	CODE	KEYS
31	f	22	GTO	81	÷
42	STK	00	0	71	x
21	DSP	34 02	RCL 2	23	LBL
04	4	34 01	RCL 1	00	0
23	LBL	81	÷	35	g
01	1	34 04	RCL 4	02	π
32	f ⁻¹	41	↑	02	2
51	SF 1	71	x	71	x
24	RTN	01	1	34 03	RCL 3
84	R/S	61	+	71	x
23	LBL	33 05	STO 5	81	÷
11	A	71	x	22	GTO
32	f ⁻¹	01	1	01	1
61	TF 1	51	-	23	LBL
33 02	STO 2	31	f	15	E
84	R/S	09	√x	31	f
33 01	STO 1	34 02	RCL 2	51	SF 1
22	GTO	81	÷	84	R/S
01	1	33 06	STO 6	35 01	g NOP
23	LBL	22	GTO	35 01	g NOP
12	B	00	0	35 01	g NOP
32	f ⁻¹	23	LBL	35 01	g NOP
61	TF 1	14	D	35 01	g NOP
33 04	STO 4	13	C	35 01	g NOP
84	R/S	34 02	RCL 2	35 01	g NOP
33 03	STO 3	34 06	RCL 6	35 01	g NOP
22	GTO	71	x	35 01	g NOP
01	1	34 04	RCL 4	35 01	g NOP
23	LBL	81	÷	35 01	g NOP
13	C	01	1	35 01	g NOP
34 04	RCL 4	61	+	35 01	g NOP
34 01	RCL 1	34 04	RCL 4		
81	÷	34 01	RCL 1		
31	f	71	x		
61	TF 1	34 05	RCL 5		

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	CODE	KEYS	CODE	KEYS
23	LBL	23	LBL	81	÷
11	A	14	D	31	f
33 01	STO 1	33	STO	09	√x
02	2	61	+	33 06	STO 6
00	0	02	2	34 05	RCL 5
09	9	33 07	STO 7	34 07	RCL 7
00	0	84	R/S	61	+
06	6	23	LBL	71	x
00	0	14	D	84	R/S
00	0	33	STO	34 06	RCL 6
00	0	61	+	34 05	RCL 5
33 05	STO 5	03	3	71	x
34 01	RCL 1	84	R/S	84	R/S
24	RTN	23	LBL	34 06	RCL 6
23	LBL	14	D	34 04	RCL 4
12	B	33 04	STO 4	34 05	RCL 5
33 01	STO 1	24	RTN	61	+
06	6	23	LBL	71	x
03	3	15	E	24	RTN
07	7	34 01	RCL 1	35 01	g NOP
08	8	32	f ⁻¹	35 01	g NOP
02	2	09	√x	35 01	g NOP
00	0	34 03	RCL 3	35 01	g NOP
00	0	34 02	RCL 2	35 01	g NOP
33 05	STO 5	51	-	35 01	g NOP
34 01	RCL 1	32	f ⁻¹	35 01	g NOP
24	RTN	09	√x	35 01	g NOP
23	LBL	51	-	35 01	g NOP
13	C	34 05	RCL 5	35 01	g NOP
33 02	STO 2	34 02	RCL 2	35 01	g NOP
24	RTN	61	+		
23	LBL	81	÷		
13	C	34 05	RCL 5		
33 03	STO 3	34 03	RCL 3		
24	RTN	61	+		

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	CODE	KEYS	CODE	KEYS
32	f ⁻¹	51	—	31	f
51	SF 1	05	5	61	TF 1
84	R/S	71	x	22	GTO
23	LBL	09	9	02	2
12	B	81	÷	61	+
41	↑	22	GTO	22	GTO
31	f	14	D	11	A
61	TF 1	23	LBL	23	LBL
22	GTO	02	2	02	2
00	0	01	1	33	STO
05	5	83	·	51	—
71	x	08	8	01	1
09	9	33	STO	23	LBL
81	÷	71	x	11	A
22	GTO	01	1	32	f ⁻¹
11	A	04	4	61	TF 1
23	LBL	05	5	22	GTO
00	0	09	9	06	6
01	1	83	·	34 01	RCCL 1
83	·	06	6	32	f ⁻¹
08	8	07	7	51	SF 1
33	STO	33	STO	24	RTN
71	x	51	—	23	LBL
01	1	01	1	06	6
22	GTO	22	GTO	33 01	STO 1
11	A	11	A	00	0
23	LBL	23	LBL	31	f
13	C	14	D	51	SF 1
41	↑	41	↑	24	RTN
31	f	02	2	35 01	g NOP
61	TF 1	07	7		
22	GTO	03	3		
02	2	83	·		
03	3	01	1		
02	2	05	5		

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	CODE	KEYS	CODE	KEYS
23	LBL	14	D	04	4
11	A	34 06	RCL 6	35 08	g R↓
00	0	33	STO	13	C
35 23	g x=y	81	÷	23	LBL
22	GTO	04	4	15	E
00	0	35 08	g R↓	03	3
35 08	g R↓	14	D	02	2
01	1	23	LBL	83	·
06	6	13	C	01	1
71	x	00	0	07	7
12	B	35 23	g x=y	04	4
23	LBL	34 04	RCL 4	33 06	STO 6
00	0	84	R/S	71	x
01	1	35 08	g R↓	00	0
06	6	33 04	STO 4	35 21	g x≠y
33	STO	00	0	35 08	g R↓
81	÷	41	↑	11	A
04	4	84	R/S	34 06	RCL 6
35 08	g R↓	23	LBL	33	STO
23	LBL	14	D	81	÷
12	B	00	0	04	4
02	2	35 23	g x=y	35 08	g R↓
08	8	22	GTO	11	A
83	·	02	2	35 01	g NOP
03	3	35 08	g R↓	35 01	g NOP
04	4	43	EEX	35 01	g NOP
09	9	03	3	35 01	g NOP
05	5	81	÷	35 01	g NOP
02	2	13	C	35 01	g NOP
03	3	23	LBL	35 01	g NOP
33 06	STO 6	02	2		
71	x	43	EEX		
00	0	03	3		
35 21	g x≠y	33	STO		
35 08	g R↓	71	x		

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	CODE	KEYS	CODE	KEYS
23	LBL	34 06	RCL 6	35 08	g R↓
11	A	33	STO	13	C
03	3	81	÷	23	LBL
83	·	05	5	15	E
07	7	35 08	g R↓	01	1
08	8	11	A	06	6
05	5	23	LBL	83	·
04	4	13	C	03	3
33 06	STO 6	00	0	08	8
71	x	35 23	g x=y	07	7
00	0	34 05	RCL 5	00	0
35 21	g x≠y	84	R/S	06	6
35 08	g R↓	35 08	g R↓	04	4
13	C	33 05	STO 5	33 06	STO 6
34 06	RCL 6	00	0	71	x
33	STO	41	↑	00	0
81	÷	84	R/S	35 21	g x≠y
05	5	23	LBL	35 08	g R↓
35 08	g R↓	14	D	14	D
13	C	00	0	34 06	RCL 6
23	LBL	35 23	g x=y	33	STO
12	B	22	GTO	81	÷
01	1	02	2	05	5
83	·	35 08	g R↓	35 08	g R↓
02	2	43	EEX	14	D
00	0	03	3	35 01	g NOP
00	0	81	÷	35 01	g NOP
09	9	13	C	35 01	g NOP
05	5	23	LBL	35 01	g NOP
33 06	STO 6	02	2	35 01	g NOP
71	x	43	EEX		
00	0	03	3		
35 21	g x≠y	33	STO		
35 08	g R↓	71	x		
11	A	05	5		

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	CODE	KEYS	CODE	KEYS
00	0	01	1	83	DSZ
33 08	STO 8	51	—	33 04	STO 4
84	R/S	43	EEX	24	RTN
23	LBL	02	2	34 03	RCL 3
11	A	71	x	34 02	RCL 2
35	g	24	RTN	34 01	RCL 1
83	DSZ	23	LBL	35	g
33 01	STO 1	02	2	05	y ^x
24	RTN	41	↑	71	x
34 04	RCL 4	41	↑	33 04	STO 4
34 03	RCL 3	43	EEX	24	RTN
81	÷	02	2	23	LBL
31	f	81	÷	15	E
07	LN	01	1	41	↑
34 02	RCL 2	61	+	01	1
31	f	33 02	STO 2	33 08	STO 8
07	LN	35 07	g x ^z y	35 07	g x ^z y
81	÷	24	RTN	24	RTN
33 01	STO 1	23	LBL	35 01	g NOP
24	RTN	13	C	35 01	g NOP
23	LBL	35	g	35 01	g NOP
12	B	83	DSZ	35 01	g NOP
35	g	33 03	STO 3	35 01	g NOP
83	DSZ	24	RTN	35 01	g NOP
22	GTO	34 04	RCL 4	35 01	g NOP
02	2	34 02	RCL 2	35 01	g NOP
34 04	RCL 4	34 01	RCL 1	35 01	g NOP
34 03	RCL 3	35	g	35 01	g NOP
81	÷	05	y ^x	35 01	g NOP
34 01	RCL 1	81	÷	35 01	g NOP
35	g	33 03	STO 3		
04	1/x	24	RTN		
35	g	23	LBL		
05	y ^x	14	D		
33 02	STO 2	35	g		

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	CODE	KEYS	CODE	KEYS
23	LBL	31	f	22	GTO
11	A	71	SF 2	04	4
33 07	STO 7	24	RTN	23	LBL
71	x	23	LBL	03	3
33 01	STO 1	01	1	34 04	RCL 4
24	RTN	32	f ⁻¹	71	x
23	LBL	51	SF 1	23	LBL
12	B	34 02	RCL 2	04	4
33 02	STO 2	34 07	RCL 7	33 08	STO 8
43	EEX	81	÷	21	DSP
02	2	34 05	RCL 5	83	*
33 05	STO 5	81	÷	02	2
24	RTN	01	1	34 05	RCL 5
23	LBL	61	+	71	x
13	C	34 01	RCL 1	83	*
32	f ⁻¹	35	g	05	5
71	SF 2	05	y ^x	61	+
31	f	33 06	STO 6	31	f
61	TF 1	01	1	83	INT
22	GTO	51	-	34 05	RCL 5
01	1	34 06	RCL 6	81	÷
33 03	STO 3	81	÷	84	R/S
24	RTN	34 07	RCL 7	35 01	g NOP
23	LBL	71	x	35 01	g NOP
14	D	34 02	RCL 2	35 01	g NOP
31	f	81	÷	35 01	g NOP
61	TF 1	34 05	RCL 5	35 01	g NOP
22	GTO	71	x	35 01	g NOP
01	1	32	f ⁻¹	35 01	g NOP
33 04	STO 4	81	TF 2	35 01	g NOP
24	RTN	22	GTO		
23	LBL	03	3		
15	E	34 03	RCL 3		
31	f	35 07	g x ² y		
51	SF 1	81	÷		

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	CODE	KEYS	CODE	KEYS
23	LBL	34 03	RCL 3	31	f
11	A	01	1	43	REG
35	g	61	+	31	f
83	DSZ	33 03	STO 3	42	STK
22	GTO	35 07	g x ² y	21	DSP
01	1	33	STO	83	*
34 01	RCL 1	61	+	02	2
34 01	RCL 1	02	2	24	RTN
34 02	RCL 2	22	GTO	23	LBL
51	-	09	9	15	E
34 04	RCL 4	23	LBL	41	↑
61	+	13	C	01	1
23	LBL	35	g	33 08	STO 8
09	9	83	DSZ	35 08	g R
84	R/S	22	GTO	24	RTN
35 07	g x ² y	03	3	35 01	g NOP
22	GTO	34 05	RCL 5	35 01	g NOP
09	9	34 04	RCL 4	35 01	g NOP
23	LBL	22	GTO	35 01	g NOP
01	1	09	9	35 01	g NOP
33 01	STO 1	23	LBL	35 01	g NOP
24	RTN	03	3	35 01	g NOP
23	LBL	41	↑	35 01	g NOP
12	B	34 05	RCL 5	35 01	g NOP
35	g	01	1	35 01	g NOP
83	DSZ	61	+	35 01	g NOP
22	GTO	33 05	STO 5	35 01	g NOP
02	2	35 07	g x ² y	35 01	g NOP
34 03	RCL 3	33	STO	35 01	g NOP
34 02	RCL 2	61	+	35 01	g NOP
22	GTO	04	4		
09	9	22	GTO		
23	LBL	09	9		
02	2	23	LBL		
41	↑	14	D		

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	CODE	KEYS	CODE	KEYS
33 03	STO 3	34 04	RCL 4	35 01	g NOP
84	R/S	15	E	35 01	g NOP
23	LBL	34 06	RCL 6	35 01	g NOP
12	B	35 07	$g x^2 y$	35 01	g NOP
33 01	STO 1	71	x	35 01	g NOP
15	E	00	0	35 01	g NOP
33 05	STO 5	35 22	$g x \leq y$	35 01	g NOP
84	R/S	22	GTO	35 01	g NOP
23	LBL	01	1	35 01	g NOP
13	C	34 02	RCL 2	35 01	g NOP
33 02	STO 2	33 01	STO 1	35 01	g NOP
15	E	34 06	RCL 6	35 01	g NOP
33 06	STO 6	33 05	STO 5	35 01	g NOP
84	R/S	22	GTO	35 01	g NOP
23	LBL	02	2	35 01	g NOP
14	D	23	LBL	35 01	g NOP
34 02	RCL 2	01	1	35 01	g NOP
34 02	RCL 2	02	2	35 01	g NOP
34 01	RCL 1	33	STO	35 01	g NOP
51	-	81	\div	35 01	g NOP
34 06	RCL 6	05	5	35 01	g NOP
34 05	RCL 5	23	LBL	35 01	g NOP
51	-	02	2	35 01	g NOP
81	\div	34 04	RCL 4	35 01	g NOP
34 06	RCL 6	33 02	STO 2	35 01	g NOP
71	x	35 00	$g LST X$	35 01	g NOP
51	-	33 06	STO 6	35 01	g NOP
33 04	STO 4	22	GTO	35 01	g NOP
51	-	14	D	35 01	g NOP
35	g	23	LBL	35 01	g NOP
06	ABS	15	E		
34 03	RCL 3	35 01	g NOP		
35 24	$g x > y$	35 01	g NOP		
34 04	RCL 4	35 01	g NOP		
84	R/S	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	CODE	KEYS	CODE	KEYS
23	LBL	71	x	35 01	g NOP
11	A	84	R/S	35 01	g NOP
33 03	STO 3	34 03	RCL 3	35 01	g NOP
35 08	$g R \downarrow$	35 07	$g x^2 y$	35 01	g NOP
42	CHS	81	\div	35 01	g NOP
33 02	STO 2	24	RTN	35 01	g NOP
35 07	$g x^2 y$	23	LBL	35 01	g NOP
41	\uparrow	13	C	35 01	g NOP
33 01	STO 1	42	CHS	35 01	g NOP
61	+	31	f	35 01	g NOP
81	\div	09	\sqrt{x}	35 01	g NOP
41	\uparrow	35 07	$g x^2 y$	35 01	g NOP
32	f^{-1}	84	R/S	35 01	g NOP
09	\sqrt{x}	35 07	$g x^2 y$	35 01	g NOP
34 03	RCL 3	24	RTN	35 01	g NOP
34 01	RCL 1	35 01	g NOP	35 01	g NOP
81	\div	35 01	g NOP	35 01	g NOP
33 03	STO 3	35 01	g NOP	35 01	g NOP
51	-	35 01	g NOP	35 01	g NOP
24	RTN	35 01	g NOP	35 01	g NOP
23	LBL	35 01	g NOP	35 01	g NOP
12	B	35 01	g NOP	35 01	g NOP
31	f	35 01	g NOP	35 01	g NOP
09	\sqrt{x}	35 01	g NOP	35 01	g NOP
35 07	$g x^2 y$	35 01	g NOP	35 01	g NOP
35	g	35 01	g NOP	35 01	g NOP
06	ABS	35 01	g NOP	35 01	g NOP
61	+	35 01	g NOP	35 01	g NOP
34 02	RCL 2	35 01	g NOP	35 01	g NOP
34 01	RCL 1	35 01	g NOP	35 01	g NOP
71	x	35 01	g NOP	35 01	g NOP
41	\uparrow	35 01	g NOP	35 01	g NOP
35	g	35 01	g NOP	35 01	g NOP
06	ABS	35 01	g NOP	35 01	g NOP
81	\div	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	CODE	KEYS	CODE	KEYS
31	f	02	2	24	RTN
43	REG	81	÷	23	LBL
00	0	33 05	STO 5	15	E
84	R/S	41	↑	33 02	STO 2
23	LBL	41	↑	35 08	g R↓
11	A	34 02	RCL 2	33 03	STO 3
41	↑	51	—	35 09	g R↑
35	g	71	x	71	x
06	ABS	34 05	RCL 5	00	0
23	LBL	34 03	RCL 3	35 24	g x>y
12	B	51	—	31	f
71	x	71	x	51	SF 1
35	g	34 05	RCL 5	34 03	RCL 3
02	π	34 04	RCL 4	41	↑
23	LBL	51	—	71	x
13	C	71	x	34 02	RCL 2
22	GTO	31	f	41	↑
06	G	09	√x	71	x
23	LBL	31	f	31	f
14	D	61	TF 1	61	TF 1
00	0	42	CHS	42	CHS
35 24	g x>y	35 01	g NOP	35 01	g NOP
31	f	32	f ⁻¹	61	+
51	SF 1	51	SF 1	35	g
35 08	g R↓	01	1	06	ABS
35	g	23	LBL	31	f
06	ABS	06	6	09	√x
33 02	STO 2	71	x	32	f ⁻¹
35 08	g R↓	33	STO	51	SF 1
33 03	STO 3	61	+	24	RTN
35 08	g R↓	01	1		
33 04	STO 4	35	g		
61	+	06	ABS		
61	+	84	R/S		
61	+	34 01	RCL 1		

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	CODE	KEYS	CODE	KEYS
23	LBL	04	4	23	LBL
11	A	81	÷	08	8
21	DSP	31	f	05	5
83	*	83	INT	05	5
00	0	35 00	g LST X	01	1
01	1	51	—	07	7
05	5	00	0	08	8
33 01	STO 1	35 21	g x≠y	84	R/S
31	f	22	GTO	23	LBL
51	SF 1	02	2	15	E
15	E	34 03	RCL 3	34 01	RCL 1
23	LBL	23	LBL	31	f
13	C	12	B	61	TF 1
31	f	33	STO	42	CHS
51	SF 1	51	—	35 01	g NOP
00	0	01	1	32	f ⁻¹
33 03	STO 3	34 01	RCL 1	51	SF 1
23	LBL	00	0	84	R/S
02	2	35 21	g x≠y	35 01	g NOP
34 03	RCL 3	22	GTO	35 01	g NOP
03	3	15	E	35 01	g NOP
35 23	g x=y	32	f ⁻¹	35 01	g NOP
01	1	61	TF 1	35 01	g NOP
12	B	22	GTO	35 01	g NOP
23	LBL	08	8	35 01	g NOP
01	1	21	DSP	35 01	g NOP
01	1	83	*	35 01	g NOP
33	STO	01	1	35 01	g NOP
61	+	03	3	35 01	g NOP
03	3	05	5	35 01	g NOP
34 01	RCL 1	00	0		
01	1	07	7		
51	—	83	*		
34 03	RCL 3	01	1		
51	—	84	R/S		

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	CODE	KEYS	CODE	KEYS
11	A	84	R/S	24	RTN
35 21	$g x \neq y$	14	D	23	LBL
00	0	35 24	$g x > y$	12	B
84	R/S	09	9	34 08	RCL 8
31	f	84	R/S	31	f
61	TF 1	15	E	51	SF 1
01	1	05	5	31	f
84	R/S	42	CHS	71	SF 2
31	f	83	.	22	GTO
81	TF 2	00	0	15	E
02	2	09	9	23	LBL
84	R/S	02	2	13	C
12	B	09	9	32	f^{-1}
35 22	$g x \leq y$	05	5	51	SF 1
03	3	08	8	32	f^{-1}
84	R/S	01	1	71	SF 2
32	f^{-1}	07	7	22	GTO
61	TF 1	08	8	15	E
04	4	43	EEX	23	LBL
84	R/S	42	CHS	14	D
32	f^{-1}	08	8	44	CLX
81	TF 2	06	6	23	LBL
05	5	31	f	15	E
84	R/S	06	TAN	35	g
13	C	21	DSP	83	DSZ
35 23	$g x = y$	09	9	35 01	g NOP
06	6	21	DSP	24	RTN
84	R/S	09	9	01	1
31	f	84	R/S	42	CHS
61	TF 1	23	LBL	84	R/S
07	7	11	A		
84	R/S	05	5		
31	f	33 08	STO 8		
81	TF 2	31	f		
08	8	42	STK		

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	CODE	KEYS	CODE	KEYS
31	f	01	R→P	33	STO
43	REG	31	f	61	+
21	DSP	03	→D.MS	06	6
09	9	32	f^{-1}	34 06	RCL 6
07	7	01	R→P	61	+
31	f	32	f^{-1}	71	x
07	LN	03	→D.MS	09	9
31	f	31	f	42	CHS
08	LOG	02	D.MS+	83	.
31	f	02	2	04	4
09	\sqrt{x}	42	CHS	08	8
31	f	35	g	00	0
04	SIN	06	ABS	04	4
31	f	35	g	07	7
05	COS	05	y^x	00	0
31	f	35	g	02	2
06	TAN	02	π	03	3
32	f^{-1}	81	÷	00	0
04	SIN	31	f	43	EEX
35	g	83	INT	42	CHS
04	$1/x$	31	f	09	9
32	f^{-1}	00	→OCT	02	2
05	COS	05	5	71	x
32	f^{-1}	35	g	32	f^{-1}
06	TAN	03	n!	83	INT
35	g	51	-	35 08	g R↓
04	$1/x$	32	f^{-1}	35 07	$g x \div y$
32	f^{-1}	00	→OCT	35 09	g R↑
07	LN	33 08	STO 8	35 01	g NOP
32	f^{-1}	35	g	84	R/S
08	LOG	83	DSZ		
32	f^{-1}	34 08	RCL 8		
09	\sqrt{x}	32	f^{-1}		
35 00	g LST X	02	D.MS+		
31	f	35 00	g LST X		

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



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