HEWLETT-PACKARD

HP-19C/HP-29C Applications Book

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.

INTRODUCTION

Welcome to the world of HP calculators. We know you will be pleased with the quality, versatility, and ease of use of your new HP-19C/HP-29C. This application book is designed to help you get the best from your calculator, whether your interest is in solving specific problems in a particular area or in learning to use the powerful programming capabilities of the HP-19C/HP-29C.

These programs have been chosen from real world problems in a variety of areas; mathematics, statistics, finance, surveying, navigation, science, medicine and games. They demonstrate the many uses of the HP-19C/HP-29C and will give you immediate calculation aids for problems you encounter every day. You will also find them useful as guides to programming techniques and models for writing your own customized software. The comments on each program listing demonstrate the approach used to reach the solution and help you follow the programmer's logic as you become an expert with your own HP-19C/HP-29C.

TABLE OF CONTENTS

A WORD ABOUT PROGRAM USAGE

This Applications Book for the HP-19C/HP-29C provides a diverse selection of programs chosen from a number of areas of interest. Each program includes a brief description, a listing of the program keystrokes, a set of instructions for using the program and one or more example problems, including the actual keystrokes required for the solution.

Explanatory comments have been incorporated in each program listing to aid your understanding of the actual working of each program. Thorough study of the commented listing can help you expand your programming repertoire since interesting techniques can often be found.

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 the data, if applicable. Data input keys consist of $\boxed{0}$ to $\boxed{9}$ and decimal point (the numeric keys), $\boxed{\text{EEX}}$ (enter exponent), and $\boxed{\text{CHS}}$ (change sign).

The KEYS column specifies the keys to be pressed after keying in the corresponding input data.

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

The following illustrates the User Instruction Form for Quadratic Equation, the first program in this book.

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	Enter coefficients of quadratic			
	x^2 coefficient	a	ENTER♦	
	x coefficient	b	ENTER♦	
	constant	c	GSB $\boxed{1}$	D
3	If D $\geqslant$ 0, roots are real		R/S	x_2
			R/S	x_1
4	If D < 0, roots are complex of		R/S	u (real part)
	form u $\pm$ iv		R/S	v
				(imaginary part)

Step 1 requires you to key in the program. Switch the HP-19C/HP-29C to PRGM mode, depress ■ CLEAR [PRGM] and key in the program steps as listed. The choice of program LABEL 1 is arbitrary and could be changed to fit the user's needs by making corresponding changes in the User's Instructions (and possibly other modifications in the program listing.) Note that some steps on the program listing require keystrokes not explicitly listed for entry in the program, e.g. LBL 1 is keyed in by three keystrokes ■ [LBL] [1]. (See the Owner's Handbook for a more detailed explanation of keying in programs.)

Step 2 of the User's Instructions asks for the coefficients of the quadratic equation. Switch the calculator to RUN mode. Coefficient a is keyed in and followed by [ENTER♦], coefficient b is keyed in and followed by [ENTER♦], and coefficient c is keyed in, followed by [GSB] [1]. D is immediately calculated and displayed and program execution stops. Upon depressing [R/S] the calculator resumes program execution, automatically determining if D is positive or negative and displaying a root of the equation. Depressing [R/S] again displays the other root.

Display of intermediate or sequential results can be accomplished in several ways; a pause may be used to display a result for approximately 1 second before resumption of program execution, or a R/S command may be used to stop execution and display the result. Execution of the program then resumes after depressing the [R/S] key. (In these programs we have usually resorted to R/S commands to eliminate the chance of missing important results during the brief pause.) Additionally, with the printer, PRINT X commands may be used to print intermediate results.

If you own the HP-19C with printer you should note that the program listings are written to provide display outputs only and do not include PRINT commands. You will want to take advantage of the printer in recording both intermediate and final results. This can best be done by substituting a PRINT X command for the R/S commands when recording sequential or intermediate results and inserting a PRINT X command at the point in the program where the final result is displayed (usually just prior to a RTN command). Use of the printer in this manner has the advantage of eliminating halts in the program due to R/S commands.

Many of the program comments show points (designated by ***) at which the PRINT X command may be inserted or substituted, if desired. If the length of the program prevents insertion of the printer commands at the various steps the results still may be recorded by manually operating [PRX] as needed.

For example, in the Quadratic Equation program the R/S instruction at step 15 could have been replaced with a PAUSE instruction if only momentary display of D was desired, or, on the HP-19C, a PRINT X command could be substituted for the R/S command at 15 and inserted after step 34 to provide a printout of the results.

QUADRATIC EQUATION

This program calculates the two roots of a quadratic equation. If the roots are real they are displayed consecutively. If complex, the real part is displayed first, followed by the imaginary part.

Equations:

The roots x_1, x_2 of

$$ax^2 + bx + c = 0$$

are given by

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

If $D = (b^2 - 4ac)/4a^2$ is positive or zero, the roots are real. In these cases, better accuracy may sometimes be obtained by first calculating the root with the larger absolute value:

If

$$-\frac{b}{2a} \geq 0, \qquad x_1 = -\frac{b}{2a} + \sqrt{D}$$

If

$$-\frac{b}{2a} < 0, \qquad x_1 = -\frac{b}{2a} - \sqrt{D}$$

In either case,

$$x_2 = \frac{c}{x_1 a}$$

If $D < 0$, the roots are complex, being

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

Remarks:

- The user merely inputs the coefficients in proper order; first a, then b, then c, being careful to observe signs for negative coefficients. The first result displayed is D. If it is positive the roots are real, if negative, they are complex.

- In the case of real roots the program tests for and calculates the larger root first for best accuracy, then displays the roots in reverse order.

01 *LBL1			
02 STO0			
03 R↓			
04 X⇄Y			
05 ST÷0	c/a		
06 ÷			
07 2			
08 ÷			
09 CHS	b/2a		
10 STO1			
11 ENT↑			
12 X²			
13 RCL0			
14 -			
15 R/S	*** D		
16 X<0?			
17 GTO0			
18 √X			
19 ST-1			
20 X⇄Y	$-b/2a - \sqrt{D}$		
21 +			
22 RCL1	$-b/2a + \sqrt{D}$		
23 LSTX			
24 X>0?			
25 R↓			
26 R↓	Select		
27 ST÷0			
28 RCL0			
29 GTO3	x_2		
30 *LBL0			
31 ABS			
32 √X			
33 X⇄Y	v		
34 *LBL3			
35 R/S	*** Display		
36 X⇄Y			
37 GTO3			

REGISTERS					
0 c/a, x_2	1 x_1	2	3	4	5
6	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	Enter coefficients of quadratic			
	x^2 coefficient	a	ENTER ↑	
	x coefficient	b	ENTER ↑	
	constant	c	GSB 1	D
3	If $D \geq 0$, roots are real		R/S	x_2
			R/S	x_1
4	If $D < 0$, roots are complex of		R/S	u (real part)
	form $u \pm iv$		R/S	v
				(imaginary part)

Example 1:

Find the roots of $x^2 + x - 6 = 0$

Keystrokes: **Outputs:**

1 ENTER ↑ 1 ENTER ↑

6 CHS GSB 1 ──────────→   6.25   (D)

R/S ────────────────────→   2.00   (x_2)

R/S ────────────────────→   −3.00   (x_1)

Example 2:

Solve the quadratic equation $2x^2 - 3x + 5 = 0$

Keystrokes: **Outputs:**

2 ENTER ↑ 3 CHS ENTER ↑

5 GSB 1 ──────────────→   −1.94   (D)

R/S ────────────────────→   0.75   (u)

R/S ────────────────────→   1.39   (v)

Since D is negative the roots are imaginary and the solutions are of the form
$x_{1,2} = 0.75 \pm 1.39i$

Example 3:

A ball is thrown straight up at a velocity of 20 meters per second from a height of 2 meters. At what time, neglecting air resistance, will it reach the ground? The acceleration of gravity is 9.81 meters/second^2. From physics:

$$f(t) = x = \frac{1}{2}gt^2 + V_0t + x_0 = 0 \quad \text{or} \quad \left(-\frac{9.81}{2}\right) t^2 + 20t + 2 = 0$$

Keystrokes: **Outputs:**

9.81 `CHS` `ENTER↑` 2 `÷` 20 `ENTER↑`
2 `GSB` `1` ─────────────────────▶   4.56   (D)
`R/S` ──────────────────────────▶   −0.10   (seconds)
`R/S` ──────────────────────────▶   4.18   (*seconds)

*The answer is 4.18 seconds. The root − 0.10 seconds is a legitimate root of the equation but is not relevant to the problem.

BASE CONVERSIONS

This program converts positive numbers to and from base 10 representations. The other base involved may be any integer from 2 to 99, inclusive.

Let x_b be the representation of the number in the original base b. Assume that it is to be converted to the representation x_B in base B. Either b or B must be 10. In general, the bases are stored manually (b in R_1, B in R_2) prior to keying in x_b and pressing [GSB] [1], which will cause the computation of x_B.

When converting numbers from base 10, b = 10. However, the number stored for b may be either 10 or 100. If the other base B < 10, then store b in R_1 as 10. If, however, B > 10, the value stored for b in R_1 should be 100.

Similarly, when numbers are converted to base 10 representations, B = 10. When b < 10, the value of B stored in R_2 should be 10; when b > 10, a value of 100 should be stored in R_2.

The table below shows examples of the four possible cases:

To convert	From Base	To Base	Store in R_1	Store in R_2
	10	2	10	2
	10	16	100	16
	2	10	2	10
	16	10	16	100

A number such as $4B6_{16}$ cannot be represented directly on the display because the display is strictly numeric. Therefore, some convention must be adopted to represent numbers R_a when a > 10. We use the convention of allocating two digit locations for each single character in R_a when a > 10.

For example, $4B6_{16}$ is represented as 041106_{16} by our convention (in hexadecimal system, A = 10, B = 11, C = 12, D = 13, E = 14, F = 15).

When displayed, this number may appear as 41106 or with an exponent

$$4.1106 \qquad 04$$

which is interpreted as $4.B6 \times 16^2$.

The displayed exponent 4 is for base 10 and only serves to locate the decimal point (in the same manner as for decimal numbers).

When base a > 10 (as in the above example), divide the displayed exponent by 2 to get the true exponent of the number. When the displayed exponent is an odd integer, shift the decimal point of the displayed number one place (to the left or right) and adjust its exponent accordingly to make the true exponent an integer.

For example, the displayed number

$$1.112 \quad -03$$

is interpreted as $B.C \times 16^{-2}$ or $0.BC \times 16^{-1}$.

Remarks:

- When the magnitude of the number is very large or very small, this program will take a long time to execute.

- The program will not give error indication for invalid inputs for x_b. For example, 981_8 will be treated the same as 1201_8.

- As the program now stands, the user is forced to make a decision at input time whether the number stored for base 10 is 10 or 100. An alternative approach would be to always store 10, never 100, and have the program decide whether to overwrite the 10 with 100 in some cases. Such an alteration of the program would require about 25 more program steps.

01 *LBL1	$x_b \to x_P$	50 GTO8	
02 STO3		51 *LBL6	
03 RCL1		52 RCL5	
04 STO5		53 RCL0	- - - - - - - - -
05 RCL2		54 Y^x	
06 STO6		55 RCL4	
07 0		56 x	
08 STO0		57 STO4	
09 STO4		58 RTN	••• x_B
10 EEX		59 *LBL7	- - - - - - - - -
11 1		60 EEX	
12 2		61 4	
13 STO0		62 +	Eliminate round-off error
14 RCL3	Shift right until < 1. R_0	63 EEX	
15 *LBL9	keeps track of no. places	64 4	
16 1	shifted (exponent) ...	65 -	
17 X>Y?		66 INT	
18 GTO8		67 RTN	
19 ST+0			- - - - - - - - -
20 CLX			
21 RCL6			
22 ÷			
23 STO3			
24 GTO9	On entry, R_3 contains		
25 *LBL8	normalized x_b:		
26 RCL6	$0 < x_b < 1$.		
27 RCL3			
28 x			
29 STO3			
30 GSB7			
31 RCL4			
32 RCL5	Build up x_B.		
33 x			
34 +			
35 STO4			
36 RCL3			
37 GSB7			
38 RCL3			
39 -			
40 ABS	Do not build mantissa		
41 STO3	beyond 10^{12}		
42 1			
43 ST-0			
44 RCL4			
45 RCL8			
46 X≤Y?	- - - - - - - - -		
47 GTO6			
48 RCL3			
49 X≠0?			

REGISTERS					
0 Used	1 b	2 B	3 x_b	4 Used	5 b
6 B	7	8 10^{12}	9	0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Store bases (one must be 10			
	or 100):			
	• Base to be converted from	b	STO 1	
	• Base to be converted to	B	STO 2	
3	Key in number in base b and			
	calculate number in base B.	x_b	GSB 1	x_B
4	For a new conversion between			
	the same bases, go to step 3;			
	to change either base, go to			
	step 2.			

Example 1:

Convert 0.2937_{10} to base 8 representation. (Since $B = 8 < 10$, $b = 10$.)

Keystrokes: **Outputs:**

10 STO 1 8 STO 2
f FIX 9 .2937 GSB 1 ⟶   0.226277543   (Base 8)

Example 2:

Convert $1.23_{10} \times 10^{-12}$ to base 16. (Since $B = 16 > 10$, $b = 100$.)

Keystrokes: **Outputs:**

100 STO 1 16 STO 2
1.23 EEX CHS 12 GSB 1 ⟶ 1.0510030 −20   (Base 16)

This is interpreted as $1.5A3_{16} \times 16^{-10}$.

Example 3:

Convert $7.200067_8 \times 8^{-10}$ to base 10. (Since $b = 8 < 10$, $B = 10$.)

Keystrokes: **Outputs:**

8 STO 1 10 STO 2
f SCI 9 7.200067
EEX CHS 10 GSB 1 ⟶   6.7522840 −09   (Base 10)

Example 4:

Convert $D.2EE4_{16} \times 16^{12}$ to base 10. (Since $b = 16 > 10$, $B = 100$.)

Keystrokes: **Outputs:**

16 STO 1 100 STO 2
13.02141404 EEX 24 GSB 1 ⟶ 3.7107314   15   (Base 10)

VECTOR OPERATIONS

This program calculates the basic vector operations of addition, dot (scalar) product, and cross product for three dimensional vectors. It also calculates the angle between two vectors. The program is capable of doing chain calculations whenever the product is a vector (refer to examples).

Equations:

Define a vector $\vec{V}$ in 3 dimensional rectangular coordinate system,

$$\vec{V} = x\,\vec{i} + y\,\vec{j} + z\,\vec{k}$$

then:

Vector addition:

$$\vec{V}_1 + \vec{V}_2 = (x_1 + x_2)\,\vec{i} + (y_1 + y_2)\,\vec{j} + (z_1 + z_2)\,\vec{k}$$

Dot or scalar product:

$$\vec{V}_1 \cdot \vec{V}_2 = x_1\,x_2 + y_1\,y_2 + z_1\,z_2$$

Cross product:

$$\vec{V}_1 \times \vec{V}_2 = (y_1 z_2 - z_1 y_2)\,\vec{i} + (z_1 x_2 - x_1 z_2)\,\vec{j} + (x_1 y_2 - y_1 x_2)\,\vec{k}$$

Angle between vectors:

$$\gamma = \cos^{-1}\frac{\vec{V}_1 \cdot \vec{V}_2}{|\vec{V}_1|\,|\vec{V}_2|}$$

Remarks:

- For two dimensional vectors, simply consider that the k component does not exist, i.e. input 0 for z's.

01 *LBL0		50 x	
02 RCL6		51 RCL3	
03 STO3		52 RCL5	
04 R↓		53 GSB8	
05 STO6		54 STO8	
06 R↓		55 R/S	
07 RCL5		56 RCL3	
08 STO2	Input $\vec{V}_1$	57 RCL4	
09 R↓		58 x	
10 STO5		59 RCL1	
11 R↓		60 RCL6	
12 RCL4		61 GSB8	
13 STO1		62 STO9	
14 R↓		63 R/S	
15 STO4		64 RCL1	$\vec{V}_1 \times \vec{V}_2$
16 RTN		65 RCL5	
17 *LBL1		66 x	
18 GSB0		67 RCL2	
19 RCL1		68 RCL4	
20 ST+4		69 *LBL8	
21 RCL4		70 x	
22 R/S		71 -	
23 RCL2	$\vec{V}_1 + \vec{V}_2$	72 STO8	
24 ST+5		73 RCL8	
25 RCL5		74 RCL9	
26 R/S		75 RCL0	
27 RCL3		76 RTN	
28 ST+6		77 *LBL4	
29 RCL6		78 GSB2	
30 RTN		79 θ	
31 *LBL2		80 ST.2	
32 GSB0		81 ST.4	
33 RCL1		82 RCL1	
34 RCL4		83 RCL4	
35 x		84 Σ+	
36 RCL2	$\vec{V}_1 \cdot \vec{V}_2$	85 RCL2	
37 RCL5		86 RCL5	
38 x		87 Σ+	
39 +		88 RCL3	
40 RCL3		89 RCL6	
41 RCL6		90 Σ+	γ
42 x		91 RC.2	
43 +		92 RC.4	
44 STO7		93 x	
45 RTN		94 √X	
46 *LBL3		95 ST÷7	
47 GSB0		96 RCL7	
48 RCL2		97 COS⁻¹	
49 RCL6		98 RTN	

REGISTERS					
0 $\vec{V}_1 \times \vec{V}_2 \, \vec{k}$	1 x_1	2 y_1	3 z_1	4 x_2	5 y_2
6 z_2	7 $\vec{V}_1 \cdot \vec{V}_2$	8 $\vec{V}_1 \times \vec{V}_2 \, i$	9 $\vec{V}_1 \times \vec{V}_2 \, \vec{j}$	.0 Used	.1 Used
.2 Used	.3 Used	.4 Used	.5 Used	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	Input the first vector $\vec{V}_1$	x_1	ENTER↑	x_1
		y_1	ENTER↑	y_1
		z_1	GSB 0	x_1
3	For vector addition, go to step 4			
	For vector dot product, go to			
	step 6.			
	For vector cross product, go			
	to step 8.			
	For the angle between two			
	vectors, go to step 10.			
4	**Vector Addition:**			
	Input the 2nd vector $\vec{V}_2$ and			
	calculate $\vec{V}_1 + \vec{V}_2$	x_2	ENTER↑	x_2
		y_2	ENTER↑	y_2
		z_2	GSB 1	$\vec{i}$
			R/S	$\vec{j}$
			R/S	$\vec{k}$
5	For a new case, go to step 2.			
6	**Vector Dot Product:**			
	Input the 2nd vector $\vec{V}_2$ and			
	calculate $\vec{V}_1 \cdot \vec{V}_2$	x_2	ENTER↑	x_2
		y_2	ENTER↑	y_2
		z_2	GSB 2	$\vec{V}_1 \cdot \vec{V}_2$
7	For a new case, go to step 2.			
8	**Vector Cross Product:**			
	Input the 2nd vector and			
	calculate $\vec{V}_1 \times \vec{V}_2$	x_2	ENTER↑	x_2
		y_2	ENTER↑	y_2
		z_2	GSB 3	$\vec{i}$
			R/S	$\vec{j}$

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
			R/S	$\vec{k}$
9	For a new case go to step 2.			
10	**Angle Between Two Vectors:**			
	Input the 2nd vector and			
	calculate γ	x_2	ENTER↑	x_2
		y_2	ENTER↑	y_2
		z_2	GSB 4	γ
11	For a new case go to step 2.			

Example 1: $\vec{V}_1 = (2, 5, 2)$, $\vec{V}_2 = (3, 3, -4)$

Addition: $\vec{V}_1 + \vec{V}_2 = (5, 8, -2)$

Keystrokes: **Outputs:**

2 ENTER↑ 5 ENTER↑
2 GSB 0 ─────────────→   2.00
3 ENTER↑ 3 ENTER↑
4 CHS GSB 1 ─────────→   5.00 $(\vec{i})$
R/S ─────────────────→   8.00 $(\vec{j})$
R/S ─────────────────→   -2.00 $(\vec{k})$

Dot product: $\vec{V}_1 \cdot \vec{V}_2 = 13.00$

Keystrokes: **Outputs:**

2 ENTER↑ 5 ENTER↑
2 GSB 0 ─────────────→   2.00
3 ENTER↑ 3 ENTER↑
4 CHS GSB 2 ─────────→   13.00 $(\vec{V}_1 \cdot \vec{V}_2)$

Cross product: $\vec{V}_1 \times \vec{V}_2 = (-26, 14, -9)$

Keystrokes: **Outputs:**

2 ENTER↑ 5 ENTER↑
2 GSB 0 ─────────────→   2.00
3 ENTER↑ 3 ENTER↑
4 CHS GSB 3 ─────────→   -26.00 $(\vec{i})$
R/S ─────────────────→   14.00 $(\vec{j})$
R/S ─────────────────→   -9.00 $(\vec{k})$

Angle:

Keystrokes:	**Outputs:**
2 `ENTER↑` 5 `ENTER↑`	
2 `GSB` `0` ——————→	2.00
3 `ENTER↑` 3 `ENTER↑`	
4 `CHS` `GSB` `4` ——————→	67.16° (γ)

Example 2:

Calculate $(\vec{V}_1 + \vec{V}_2) \cdot \vec{V}_3$ for $\vec{V}_1 = (1.10, 3.00, 4.40)$
$\vec{V}_2 = (1.24, 2.17, 3.03)$, and $\vec{V}_3 = (0.072, 0.231, 0.409)$

Keystrokes:	**Outputs:**
1.10 `ENTER↑` 3 `ENTER↑`	
4.40 `GSB` `0` ——————→	1.10
1.24 `ENTER↑` 2.17 `ENTER↑`	
3.03 `GSB` `1` ——————→	2.34 ⎫
`R/S` ——————→	5.17 ⎬ $(\vec{V}_1 + \vec{V}_2)$
`R/S` ——————→	7.43 ⎭
0.072 `ENTER↑` 0.231 `ENTER↑`	
0.409 `GSB` `2` ——————→	4.40 $((\vec{V}_1 + \vec{V}_2) \cdot \vec{V}_3)$

COMPLEX OPERATIONS

This program allows for chained calculations involving complex variables. The four operations of complex arithmetic ($+$, $-$, $\times$, $\div$) are provided, as well as several of the most used functions of a complex variable z($|z|$, z^n, and $z^{1/n}$). Functions and operations may be mixed in the course of a calculation to allow evaluation of expressions like $z_3/(z_1 + z_2)$, $|z_1 + z_2|$, ..., etc., where z_1, z_2 and z_3 are complex numbers of the form $x + iy$.

Arithmetic Operations

An arithmetic operation needs two numbers to operate on. Both numbers must be input before the operation can be performed. Suppose that $z_1 = 2 + 3i$, $z_2 = 5 - i$, and we wish to find $z_1 - z_2$. This can be calculated by the keystrokes:

2 ENTER↑ 3 GSB 0 5 ENTER↑ 1 CHS GSB 2

The result $z_3 = u + iv$ is found to be $-3 + 4i$. This result is now stored by the program in place of the second complex number z_2. A further calculation $z_3 \times z_4$ could be performed by inputting z_4 and depressing GSB 3 for multiplication. This type of chaining can be continued indefinitely, and functions can be interspersed with arithmetic operations.

Equations:

$$\text{Let } z_j = x_j + iy_j = r_j e^{i\theta_j}, \; j = 1, 2$$

$$z = x + iy = r e^{i\theta}$$

Where
$$r = \sqrt{x^2 + y^2}$$

Let the result in each case be $u + iv$

$$z_1 + z_2 = u + iv = (x_1 + x_2) + i(y_1 + y_2)$$

$$z_1 - z_2 = u + iv = (x_1 - x_2) + i(y_1 - y_2)$$

$$z_1 \cdot z_2 = r_1 \cdot r_2 \cdot e^{i(\theta_1 + \theta_2)} = u + iv$$

$$z_1/z_2 = \frac{r_1}{r_2} e^{i(\theta_1 - \theta_2)} = u + iv$$

$$|z| = r = \sqrt{x^2 + y^2}$$

$$z^n = r^n e^{in\theta} \qquad n = \pm(1, 2, 3, ...)$$

$$z^{1/n} = r^{1/n} e^{i\left(\frac{\theta}{n} + \frac{360k}{n}\right)}, \; k = 0, 1, ..., n-1$$

01 *LBL0		50 GTO8	
02 RCL4		51 *LBL9	
03 STO2		52 RCL2	
04 R↓		53 RCL1	
05 STO4		54 →P	
06 R↓		55 STO5	
07 RCL3	Input z_1	56 X≷Y	
08 STO1		57 STO6	
09 R↓		58 *LBL5	
10 STO3		59 RCL4	$|z_1|$
11 0		60 RCL3	
12 STO0		61 →P	
13 RTN		62 RTN	
14 *LBL2		63 *LBL6	
15 CHS		64 STO7	
16 X≷Y	$z_1 - z_2$	65 GSB5	z^n
17 CHS		66 RCL7	
18 X≷Y		67 Y^x	
19 *LBL1		68 STO5	
20 GSB0		69 X≷Y	
21 RCL1		70 RCL7	
22 ST+3		71 x	
23 RCL2	$z_1 + z_2$	72 STO6	
24 ST+4		73 GTO8	
25 RCL3		74 *LBL7	
26 R/S		75 STO7	
27 RCL4		76 GSB5	
28 R/S		77 RCL7	
29 *LBL3		78 1/X	
30 GSB0		79 Y^x	
31 GSB9		80 X≷Y	
32 ST×5		81 RCL7	
33 X≷Y	$z_1 \times z_2$	82 ÷	
34 ST+6		83 3	
35 *LBL8		84 6	$z^{1/n}$
36 RCL6		85 0	
37 RCL5		86 RCL0	
38 →R		87 x	
39 STO3		88 RCL7	
40 R/S		89 ÷	
41 X≷Y		90 +	
42 STO4		91 X≷Y	
43 RTN		92 →R	
44 *LBL4		93 R/S	
45 GSB0		94 X≷Y	
46 GSB9		95 R/S	
47 ST÷5	z_1/z_2	96 ISZ	
48 X≷Y		97 RCL7	
49 ST-6		98 GTO7	

REGISTERS

0 K	1 x_1	2 y_1	3 x_2, Last x	4 y_2, Last y	5 $x, \div, r^n$
6 +, −, $n\theta$	7 n	8	9	.0	.1
.2	.3	.4	5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Key in the first complex number.			
	$z_1 = x_1 + iy_1$	x_1	ENTER♦	
		y_1	GSB 0	0
3	For a function, go to step 7, for			
	arithmetic, go to step 4. A com-			
	plex result is $u + iv$			
4	**Arithmetic**			
	Key in the second complex			
	number $z_2 = x_2 + iy_2$	x_2	ENTER♦	
		y_2		
5	Select one of the four:			
	• Add (+)		GSB 1	u
			R/S	v
	• Subtract (−)		GSB 2	u
			R/S	v
	• Multiply (×)		GSB 3	u
			R/S	v
	• Divide (÷)		GSB 4	u
			R/S	v
6	The result of the operation has			
	been stored, go to step 7 for a			
	function or to step 4 for further			
	arithmetic.			
7	**Functions**			
	Select one of the 3 functions:			
	• Magnitude ($\lvert z_1 \rvert$)		GSB 5	$\lvert z \rvert$
	• Raise z to integer power (z_1^n)	n	GSB 6	u
			R/S	v
	• Find the roots of ($z^{1/n}$)	n	GSB 7	u
			R/S	v

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
			R/S	u_2
			R/S	v_2
			⋮	⋮
			R/S	u_n
			R/S	v_n
8	The result, if complex, has been calculated; go to step 4 for arithmetic or to step 7 for another function.			

Examples:

Keystrokes: **Outputs:**

1. $(3+4i) + (7.4 - 5.6i) = 10.40 - 1.60i$

    3 ENTER↑ 4 GSB [0] 7.4 ENTER↑ 5.6 CHS GSB [1] ⟶   10.40

    R/S ⟶   −1.60

2. $(3+4i) - (7.4 - 5.6i) = -4.40 + 9.60i$

    3 ENTER↑ 4 GSB [0] 7.4 ENTER↑ 5.6 CHS GSB [2] ⟶   −4.40

    R/S ⟶   9.60

3. $(3.1+4.6i)(5-12i) = 70.70 - 14.20i$

    3.1 ENTER↑ 4.6 GSB [0] 5 ENTER↑

    12 CHS GSB [3] ⟶   70.70

    R/S ⟶   −14.20

4. $\dfrac{3+4i}{7-2i} = 0.25 + 0.64i$

    3 ENTER↑ 4 GSB [0] 7 ENTER↑ 2 CHS GSB [4] ⟶   0.25

    R/S ⟶   0.64

5. $\dfrac{1}{2+3i} = 0.15 - 0.23i$

    1 ENTER↑ 0 GSB [0] 2 ENTER↑ 3 GSB [4] ⟶   0.15

    R/S ⟶   −0.23

    OR:

    2 ENTER↑ 3 GSB [0] 1 CHS GSB [6] ⟶   0.15

    R/S ⟶   −0.23

6. $(7-2i)^2 = 45.00 - 28.00i$

    7 `ENTER↑` 2 `CHS` `GSB` `0` 2 `GSB` `6` —————————→ 45.00

    `R/S` ———————————————————————→ −28.00

7. $\sqrt{7+6i} = \pm (2.85 + 1.05i)$

    7 `ENTER↑` 6 `GSB` `0` 2 `GSB` `7` —————————→ 2.85

    `R/S` ———————————————————————→ 1.05

    `R/S` ———————————————————————→ −2.85

    `R/S` ———————————————————————→ −1.05

8. $\dfrac{23+13i}{(-2+i)+(4-3i)} = 2.50 + 9.00i$

    2 `CHS` `ENTER↑` 1 `GSB` `0` 4 `ENTER↑` 3 `CHS` `GSB` `1` —→ 2.00

    `R/S` ———————————————————————→ −2.00

    1 `CHS` `GSB` `6` ———————————————→ 0.25

    ———————————————————————————→ 0.25

    `R/S` ———————————————————————→ 2.50

    23 `ENTER↑` 13 `GSB` `3` —————————————→ 2.50

    `R/S` ———————————————————————→ 9.00

SYSTEM OF LINEAR EQUATIONS WITH 3 UNKNOWNS

This program uses Cramer's rule to solve systems of linear equations with three unknowns.

Equations:

A system of linear equations can be expressed as

$$A\bar{x} = \bar{b}$$

For 3 Unknowns,
$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \bar{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Determinant of the system

$$Det = a_{11}(a_{22}\,a_{33} - a_{23}\,a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

$$b_i\text{'s are solved by } b_i = \frac{\det\,(i)}{Det}$$

Where det (i) is the determinant of the A matrix with the ith column replaced by $\bar{b}$.

Remarks:

If "Error" occurs while running the program, then possibly the determinant is zero. i.e. the system is linearly dependent and this program is not applicable.

01 *LBL1		50 STO0	
02 0		51 GSB0	
03 STO0		52 GSB1	
04 RCL6		53 RC.5	
05 RCL8	Input a_{ij}'s and calculate	54 ÷	
06 GSB9	Det	55 ST.0	- - - - - - - - -
07 RCL4		56 RC.4	
08 RCL9		57 STO0	
09 GSB9		58 GSB0	
10 RCL5		59 RC.0	Swap register contents
11 RCL7		60 RTN	
12 GSB9		61 *LBL0	
13 CHS		62 RCL i	
14 RCL3		63 RC.1	
15 RCL8		64 STO i	
16 GSB9		65 X⇄Y	
17 RCL1		66 ST.1	
18 RCL9		67 RCL0	
19 GSB9		68 3	
20 RCL2		69 +	
21 RCL7		70 STO0	
22 *LBL9		71 RCL i	
23 DSZ		72 RC.2	
24 RCL i	- - - - - - - - -	73 STO i	
25 x		74 X⇄Y	
26 x		75 ST.2	
27 +		76 RCL0	
28 ST.0	Input b_1, b_2, b_3 and	77 3	
29 RTN	calculate x_1, x_2, and x_3	78 +	
30 *LBL2		79 STO0	
31 ST.3		80 RCL i	
32 R↓		81 RC.3	
33 ST.2		82 STO i	
34 R↓		83 X⇄Y	
35 ST.1		84 ST.3	
36 GSB1		85 RTN	
37 RC.0			
38 ST.5			
39 1			
40 GSB7			
41 R/S	***		
42 2	- - - - - - - - -		
43 GSB7			
44 R/S	***		
45 3			
46 GSB7			
47 R/S	***		
48 *LBL7	Subroutine to calculate		
49 ST.4	det (i)		

REGISTERS					
0 Index	1 a_{11}	2 a_{12}	3 a_{13}	4 a_{21}	5 a_{22}
6 a_{23}	7 a_{31}	8 a_{32}	9 a_{33}	.0 Det	.1 b_1
.2 b_2	.3 b_3	.4 Index	.5 Det	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	Store elements of A matrix	a_{11}	STO 1	a_{11}
		a_{12}	STO 2	a_{12}
		a_{13}	STO 3	a_{13}
		a_{21}	STO 4	a_{21}
		a_{22}	STO 5	a_{22}
		a_{23}	STO 6	a_{23}
		a_{31}	STO 7	a_{31}
		a_{32}	STO 8	a_{32}
		a_{33}	STO 9	a_{33}
3	(Optional) to calculate determinant		GSB 1	Det
4	Input $\bar{b}$ to calculate $\bar{x}$	b_1	ENTER ↑	b_1
		b_2	ENTER ↑	b_2
		b_3	GSB 2	x_1
			R/S	x_2
			R/S	x_3
5	For a new b with the same system, go to step 4.			
6	For a new system, go to step 2.			

Example:

Find x_1, x_2, and x_3 for the following system.

$$\begin{bmatrix} 19 & -4 & -15 \\ -4 & 22 & -10 \\ -15 & -10 & 26 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix}$$

Keystrokes: **Outputs:**

19 STO 1 , 4 CHS STO 2 ,
15 CHS STO 3 , 4 CHS STO 4 ,
22 STO 5 , 10 CHS STO 6 ,
15 CHS STO 7 , 10 CHS STO 8 ,
26 STO 9
GSB 1 ⟶   2402.00   (Det)
40 ENTER↑ 0 ENTER↑
0 GSB 2 ⟶   7.86   (x_1)
R/S ⟶   4.23   (x_2)
R/S ⟶   6.16   (x_3)

ANNUITIES AND COMPOUND AMOUNTS

These programs (1st part and 2nd part) can be used to solve a variety of problems involving money, time and interest. The following variables can be inputs or outputs:

n, which is the number of compounding periods. (For a 30 year loan with monthly payments, n $= 12 \times 30 = 360$.)

i, which is the periodic interest rate expressed as a percent. (For other than annual compounding, divide the annual percentage rate by the number of compounding periods in a year, i.e. 8% annual interest compounded monthly equals 8/12 or 0.667%.)

PMT, which is the periodic payment.

PV, which is the present value of the cash flows or compound amounts.

FV, which is the future value of a compounded amount or a series of cash flows.

BAL, which is the balloon or remaining balance at the end of a series of payments.

Accumulated interest and remaining balance may also be computed with this program.

The program accommodates payments which are made at the end of compounding periods or at the beginning. Payments made at the end of compounding periods (ordinary annuity) are common in direct reduction loans and mortgages while payments at the beginning of compounding periods (annuity due) are common in leasing.

This program uses the convention that cash outlays are input as negative, and cash incomes are input as positive.

1st part: When i is known

The initialization (**GSB** [0]) performs two functions:

1. It sets PMT, PV, and BAL to zero (n and i are not affected).
2. It toggles for the ordinary annuity mode (display $= 1$), and annuity due mode (display $= 0$).

Pressing **GSB** [0] provides a safe, convenient, easy to remember method of preparing the calculator for a new problem. It is not necessary to use **GSB** [0] between problems containing the same combination of variables. For instance, any number of n, i, PMT, FV problems involving different numbers and/or different combinations of knowns could be done in succession without re-initializing. Only the values which change from problem to problem would have to be keyed in. To change the combination of variables without using

GSB **0** , simply input zero for any variable which is no longer applicable. To go from n, i, PMT, PV problems to n, i, PV, FV problems, a zero would be stored (0 **STO** **3**) in place of PMT. Table I summarizes these procedures.

2nd part: Solving for i
Newton's method is applied to solve problems with unknown i. (Refer to page 81: Newton's Method-Solution to $f(x) = 0$).

Table I
Possible Solutions Using *Annuities and Compound Amounts*

Allowable Combination of Variables	Applications		Initial Procedure
	Ordinary Annuity	Annuity Due	
n, i, PMT, PV(Input any three and calculate the fourth.)	Direct reduction loan Discounted notes Mortgages	Leases	Use **GSB** **0** or set BAL to zero
n, i, PMT, PV, BAL. (Input any four and calculate the fifth.)	Direct reduction loan with balloon Discounted notes with balloon	Leases with residual values	None
n, i, PMT, FV (Input any three and calculate the fourth.)	Sinking fund	Periodic savings insurance	Use **GSB** **0** or set PV to zero
n, i, PV, FV (Input any three and calculate the fourth.)	Compound amount Savings (Annuity mode is not applicable and has no effect)		Use **GSB** **0** or set PMT to zero.

Equations:

$$PV = \frac{PMT}{i} \; A\left[1 - (1 + i)^{-n}\right] + (BAL \; or \; FV)(1 + i)^{-n}$$

where
$$A = \begin{cases} 1 & \text{ordinary annuity} \\ (1 + i) & \text{annuity due.} \end{cases}$$

Remarks:
- The equation above is solved for i using Newton's method where:

$$i_n = i_{n-1} - \frac{f(i_{n-1})}{f'(i_{n-1})}$$

This is why solutions involving PMT and i take longer than other solutions. It is quite possible to define problems which cannot be solved by this technique. Such problems usually result in an error message but may simply continue to run indefinitely.

- Interest problems with balloon payment of opposite sign to the periodic payments may have more than one mathematically correct answer (or no answer at all). While this program may find one of the answers, it has no way of finding or indicating other possibilities.

1st Part: When i Is Known

Program		Program	
01 *LBL0		50 +	
02 CLX		51 CHS	
03 STO3		52 STO4	
04 STO4	Toggling	53 RTN	***
05 STO5		54 *LBL5	
06 RC.2	1 for ordinary annuity	55 GSB9	
07 X=0?	0 for annuity due	56 RCL4	
08 GTO8		57 +	
09 0		58 RCL8	Calculate
10 ST.2		59 ÷	FV (BAL)
11 RTN		60 CHS	
12 *LBL8		61 STO5	
13 1		62 RTN	***
14 ST.2		63 *LBL9	
15 RTN		64 1	
16 *LBL1		65 ST.1	
17 0		66 RCL2	
18 STO1		67 %	
19 GSB9		68 STO9	
20 RCL5		69 +	Calculate
21 LSTX	Calculate n	70 STO7	
22 -		71 RC.2	$\dfrac{PMT}{i}\,[1-(1+i)^{-n}]\;R_1$
23 RCL4		72 X=0?	
24 Pi		73 X≷Y	
25 +		74 ST.1	
26 ÷		75 RCL7	
27 CHS		76 RCL1	
28 LN		77 CHS	
29 RCL7		78 Y^x	
30 LN		79 STO8	
31 ÷		80 RCL5	
32 STO1		81 x	
33 RTN	***	82 ST.3	
34 *LBL3		83 1	
35 1		84 RCL8	
36 STO3		85 -	
37 GSB9	Calculate PMT	86 ST.0	
38 1/X		87 RCL3	
39 RCL4		88 RCL9	
40 RC.3		89 ÷	
41 +		90 STO0	
42 x		91 RC.1	
43 CHS		92 x	
44 STO3		93 x	
45 RTN	***	94 RTN	
46 *LBL4	Calculate PV		
47 1			
48 STO4			
49 GSB9			

REGISTERS					
0 PMT/i	1 n	2 i	3 PMT	4 PV	5 FV (BAL)
6 $n(1+i)^{n-1}$	7 $1+i$	8 $(1+i)^{-n}$	9 i/100	.0 $1-(1+i)^{-n}$	.1 1 or $1+i$
.2 annuity flag	3 Used	4	5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

2nd Part: Solving For i

01 *LBL1	Annuity	50 –	
02 1		51 ÷	Calculate next i
03 GTO3	— — — — — — — — — —	52 CHS	
04 *LBL2		53 GSB9	— — — — — — — — — —
05 0	Annuity due	54 RCL2	
06 *LBL3		55 ÷	
07 ST.2	— — — — — — — — — —	56 ABS	
08 0	Clear R_2 for sum of	57 RC.5	
09 STO2		58 X≤Y?	
10 RCL5	i terms	59 GTO8	Test increment to i for
11 RCL4		60 RCL2	limit
12 +	— — — — — — — — — —	61 RTN	
13 RCL1	$\dfrac{nPMT + BAL + PV}{n}$	62 *LBL9	
14 ÷		63 EEX	— — — — — — — — — —
15 RCL3		64 2	
16 +	— — — — — — — — — —	65 ×	Calculate i to % and add
17 RCL4		66 ST+2	to i
18 ÷	guess for i	67 RTN	
19 CHS	— — — — — — — — — —	68 *LBL0	
20 .		69 1	— — — — — — — — — —
21 9		70 ST.1	
22 CHS	If guess is less than –0.9,	71 RCL2	
23 X≤Y?	use –0.9 for guess.	72 %	
24 X⇄Y		73 STO9	
25 GSB9		74 +	1 to R_1 for ordinary
26 X=0?	— — — — — — — — — —	75 STO7	annuity
27 RTN		76 RC.2	
28 *LBL8		77 X=0?	
29 GSB0	Calculate f(i)	78 X⇄Y	Calculate
30 +		79 ST.1	$\dfrac{PMT}{i}\,[1 - (1+i)^{-n}] \times R_1$
31 RCL4	— — — — — — — — — —	80 RCL7	
32 +		81 RCL1	
33 RCL8		82 CHS	
34 RCL1		83 Y˟	
35 RCL7	Calculate f'(i)	84 STO8	
36 ÷		85 RCL5	
37 ×		86 ×	
38 STO6		87 1	
39 RC.0		88 RCL8	
40 RCL9		89 –	
41 ÷		90 ST.0	
42 –		91 RCL3	
43 RC.1		92 RCL9	
44 ×		93 ÷	
45 RCL0		94 STO0	
46 ×		95 RC.1	
47 RCL6		96 ×	
48 RCL5		97 ×	
49 ×	f(i)/f'(i)	98 RTN	

REGISTERS					
0 PMT/i	1 n	2 i	3 PMT	4 PV	5 FV(BAL)
6 $n(1+i)^{-n-1}$	7 $1+i$	8 $(1+i)^{-n}$	9 i/100	.0 $1-(1+i)^{-n}$	.1 1 or $1+i$
.2 annuity flag	.3 used	4	.5 10^{-6}	16	17
18	19	20	21	22	23
24	25	26	27	28	29

1st Part: When i Is Known

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Toggling for ordinary annuity			
	(1.00) and annuity due (0.00)		GSB 0	1.00/0.00
3	Input the known values (i must			
	be known):			
	Number of periods	n	STO 1	n
	Periodic interest rate	i (%)	STO 2	i (%)
	Periodic payment	PMT	STO 3	PMT
	Present value	PV	STO 4	PV
	Future value, balloon or			
	balance	FV (BAL)	STO 5	FV, (BAL)
4	Calculate the unknown value			
	Number of periods		GSB 1	n
	Periodic payment		GSB 3	PMT
	Present value		GSB 4	PV
	Future value, balloon or			
	balance		GSB 5	FV, (BAL)
6	For a new case, go to step 3			
	and change appropriate values.			
	Input zero for any value not			
	applicable in the new case.			

2nd Part: Solving For i

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input the known values: (0 for			
	not existing values)			
	Number of periods	n	STO 1	n
	Periodic payment	PMT	STO 3	PMT
	Present value	PV	STO 4	PV

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
	Future value, balloon or			
	balance	FV, (BAL)	`STO` `5`	FV, (BAL)
	and the tolerance for i (say			
	$\epsilon = 10^{-6}$)	ϵ	`STO` `.` `5`	ϵ
3	Calculate interest rate.			
	For ordinary annuity		`GSB` `1`	i (%)
	For annuity due		`GSB` `2`	i (%)
4	For a new case, go to step 2.			

Example 1:

If you place $155 in a savings account paying 5¾% compounded monthly, what sum of money may you withdraw at the end of 9 years?
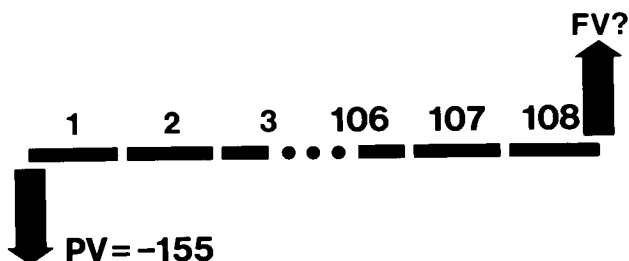
Keystrokes: **Outputs:**

(Key in the 1st program)

`GSB` `0` ────────────────▶   1.00   (ordinary annuity)

9`ENTER↑` 12`×` `STO` `1` ────▶   108.00   (# of month
                                                compounding)

5.75`ENTER↑` 12`÷` `STO` `2` ──▶   0.48   (% monthly
                                                interest rate)

155`CHS` `STO` `4` ─────────────▶   −155.00 (cash outlay)

`GSB` `5` ───────────────────▶   259.74 (FV)

If the interest is changed to 6% what is the sum?

6`ENTER↑` 12`÷` `STO` `2` ─────▶   0.50   (% monthly
                                                interest rate)

`GSB` `5` ───────────────────▶   265.62 (FV)

Example 2:

You receive $30000 from the bank as a 30 year, 9% mortgage. What monthly payment must you make to the bank to fully amortize the mortgage?
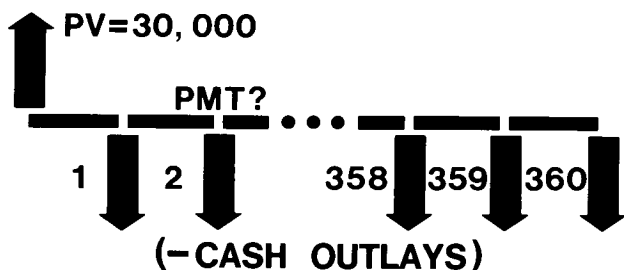
PV=30, 000

PMT? • • •

1   2   358 359 360

(−CASH OUTLAYS)

Keystrokes:	**Outputs:**	
(Key in the 1st program)		
GSB [0] ⟶	1.00	
30 ENTER↑ 12 [×] STO [1] ⟶	360.00	(# monthly payments)
9 ENTER↑ 12 [÷] STO [2] ⟶	0.75	(% monthly interest rate)
30000 STO [4] ⟶	30000.00	(PV)
GSB [3] ⟶	−241.39	(PMT)

Example 3:

Two individuals are constructing a loan with a balloon payment. The loan amount is $3,600 and it is agreed that the annual interest rate will be 10% with 36 monthly payments of $100. What balloon payment amount, to be paid coincident with the 36th payment, is required to fulfill the loan agreement?

(Note the cash flow diagram below is with respect to the loaner. For the loanee, the appropriate diagram will be exactly the opposite.)

1   2   3   34 35 36

PMT 100 PMT 100   PMT 100 PMT 100 PMT 100

BAL?

PV = −3600
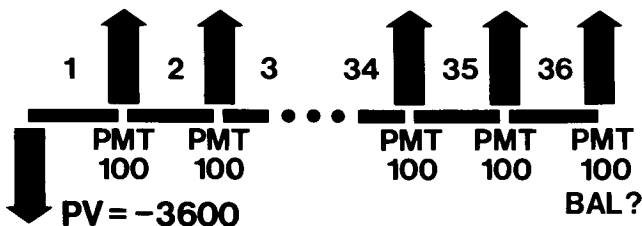
Keystrokes: **Outputs:**

(Key in the 1st program)

GSB 0 ──────────────► 1.00

36 STO 1 10 ENTER↑ 12 ÷

STO 2 100 STO 3 3600

CHS STO 4 GSB 5 ──────► 675.27

(Note that the final payment is $675.27 + $100.00 = $775.27 since the final payment falls at the end of the last period.)

Example 4:

A corporation has determined that a certain piece of equipment costing $50,000 will be required in 3 years. Assuming a fund paying 7% compounded quarterly is available, what quarterly payment must be made in order to withdraw this cost from the fund if savings are to start at the end of this quarter?
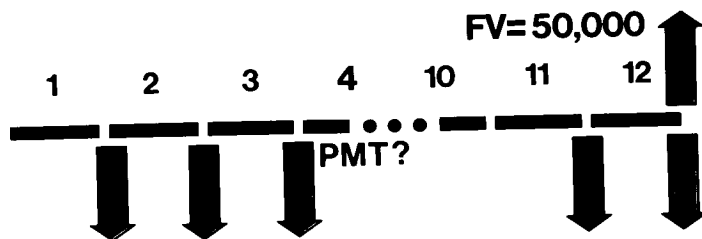
Keystrokes: **Outputs:**

(Key in the 1st program)

GSB 0 ──────────────► 1.00

3 ENTER↑ 4 × STO 1 7 ENTER↑

4 ÷ STO 2 50000 STO 5

GSB 3 ──────────────► −3780.69

What single amount, invested immediately, would provide the same effect?

0 STO 3 GSB 4 ──────► −40602.89

Example 5:

This program may also be used to calculate accumulated interest/remaining balance for loans. The accumulated interest between two points in time, n_1 and n_2, is just the total payments made in that period less the principal reduction in that period. The principal reduction is the difference of the remaining balances for the two points in time. The following example demonstrates the concepts above.

For a 360 month, $50,000 loan at 9½% annual interest, find the remaining balance after the 24th payment and the accrued interest for payments 13-24 (between the 12th and 24th payments!).

First we must calculate the payment on the loan:

Keystrokes: **Outputs:**

(Key in the 1st program)
GSB 0 ───────────────►   1.00

360 STO 1 9.5 ENTER▴ 12 ÷
STO 2 50000 CHS STO 4
GSB 3 ───────────────►   420.43   (payment)

The remaining balance is found:
24 STO 1 GSB 5 ──────────►   49352.76   (remaining balance at month 24)

Store this remaining balance and calculate the remaining balance at period 12:
STO • 4 12 STO 1
GSB 5 ───────────────►   49691.68

The principal reduction between payments 12 and 24 is:
RCL • 4 ─ ───────────►   338.92

The accrued interest is 12 payments less the principal reduction:
RCL 3 12 × ─────────►   5045.13   (total paid out)
x⇆y ─ ───────────────►   4706.20   (accrued interest)

Example 6:

A "third" party leasing firm is considering the purchase of a mini-computer priced at $63,000 and intends to achieve a certain annual yield by leasing the computer to a customer for a 5-year period. Ownership is retained by the leasing firm and at the end of lease they expect to be able to sell the equipment for at least $10,000. If the monthly payment is $1300.16, what is the annual yield? (Since lease payments occur at the start of the periods, this is an annuity due problem).

PV = −63,000
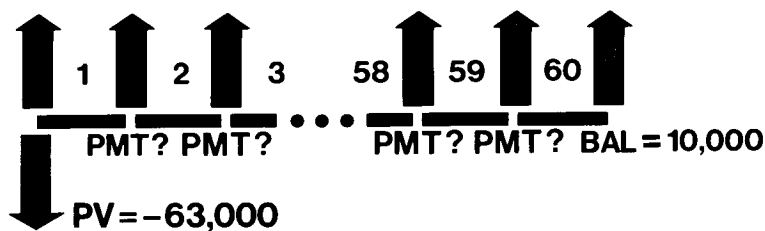
Keystrokes: **Outputs:**

(Key in the 2ⁿᵈ program)

5 ENTER↓ 12 × STO 1,
1300.16 STO 3 63000 CHS
STO 4 10000 STO 5 EEX
CHS 6 STO · 5 GSB 2 ———→   1.08   (% per month)
12 × ———————————————→   13.00   (% per year)

Example 7:

A fixed term annuity is available which requires a $35,000 initial deposit. In return the depositor will receive monthly payments of $231 for 20 years. What annual interest rate is being applied?

PV = −35,000
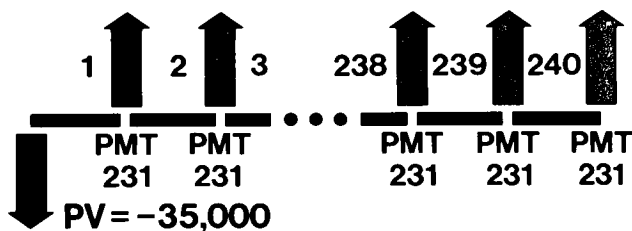
Keystrokes: **Outputs:**

(Key in the 2ⁿᵈ program)
20 ENTER↓ 12 × STO 1 ———→   240.00   (# monthly
                                             payments)

231 STO 3 ——————————→   231.00   (monthly income)

35000 CHS STO 4 ————→   −35000.00 (initial cash
                                             deposit)

0 STO 5 ——————————————→   0.00   (FV = 0)

EEX CHS 6 STO · 5 ———→   1. −06   (ε)

GSB 1 ————————————————→   0.42   (0.42% monthly)

12 × ————————————————→   5.00   (5% annual
                                             interest rate)

Example 8:

Suppose you deposit $100 today in the bank, after 3 years you will have a total of $116.08. If the interest is compounded quarterly, what is the interest rate?

Keystrokes: **Outputs:**

(Key in the 2nd program)
3 ENTER↑ 4 ⊗ STO 1 0 STO 3
100 CHS STO 4 116.08 STO 5
EEX CHS 6 STO · 5
GSB 1 ————————————→   1.25   (% quarter)
4 ⊗ ————————————————→   5.00   (% annual)

DISCOUNTED CASH FLOW ANALYSIS
NET PRESENT VALUE

Assuming a minimum desired yield (cost of capital, discount rate), this program finds the present value of the future cash flows generated by the investment and subtracts the initial investment from this amount. If the final net present value is a positive value, the investment exceeds the profit objectives assumed. If the final net present value is a negative value, then the investment is not profitable to the extent of the desired yield. If the net present value is zero, the investment meets the profit objectives.

The function associated with the **GSB** **3** key (#) is designed to accommodate those situations where a series of the cash flows are equal. You enter the number of times these equal periodic cash flows occur with **GSB** **3** , and then the amount only once with **GSB** **4** . The program automatically assumes 1 for #. If the cash flow occurs only once, there is no need to enter anything for #.

Zero must be entered for all periods with no cash flow. When a cash flow other than the initial investment is an outlay (additional investment, loss, etc.) the value must be entered as a negative number with **CHS** .

Cash flows are assumed to occur at the end of cash flow periods.

Equation:

$$NPV_k = -INV + \sum_{k=1}^{n} \frac{CF_k}{(1 + i)^k}$$

where:

    n = number of cash flows

    CF_k = k^{th} cash flow

    NPV_k = net present value after k^{th} cash flow

01 *LBL1		50 R↓	***
02 CHS		51 RTN	Recall Σn
03 STO1		52 *LBL5	
04 0	−NPV → R₁	53 RCL9	
05 STO9	0 → R₉	54 RTN	
06 1	1 → R₃		
07 STO3			
08 RCL1			
09 CHS			
10 RTN			
11 *LBL2			
12 EEX			
13 2			
14 ÷	$\frac{i}{100} \to R_2$		
15 STO2			
16 LSTX			
17 x			
18 RTN			
19 *LBL3			
20 STO3	# → R₃		
21 RTN			
22 *LBL4			
23 STO4			
24 1			
25 RCL2			
26 +			
27 RCL3			
28 ST+9	Calculate present value of		
29 Yˣ	series		
30 STO5			
31 RCL1			
32 x			
33 RCL5			
34 1			
35 −			
36 RCL2			
37 ÷			
38 RCL4			
39 x			
40 +			
41 STO1			
42 1			
43 RCL2			
44 +			
45 RCL9			
46 Yˣ			
47 ÷	Reset n to 1		
48 1			
49 STO3			

REGISTERS

0	1 NPV	2 i/100	3 #	4 CF	5 $(1 + i)^n$
6	7	8	9 Σn	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input:			
	Initial investment amount	INV	GSB 1	INV
	Periodic interest (discount)			
	rate	i (%)	GSB 2	i (%)
3	Input the number of equal cash			
	flows if greater than 1.	#	GSB 3	#
4	Input cash flow amounts and			
	calculate net present value	CF	GSB 4	NPV
5	(Optional): Display total number			
	of cash flows entered so far.		GSB 5	n
6	For next cash flow go to step 3.			
7	For a new case go to step 2.			

Example 1:

An investor has an opportunity to purchase a piece of property for $70,000. If the going rate of return on this type of investment is 13.75%, and the after-tax cash flows are forecast as follows, should the investor purchase the property?

Year	Cash Flow ($)
1	$14,000
2	11,000
3	10,000
4	10,000
5	10,000
6	9,100
7	9,000
8	9,000
9	4,500
10	71,000 (property sold in 10th year)

Keystrokes:

70000 GSB 1 13.75 GSB 2
14000 GSB 4 ⟶

Outputs:

−57692.31   (NPV after 1 cash flow)

11000 GSB 4 ————————→	–49190.92	(NPV after 2 cash flows)
3 GSB 3 10000 GSB 4 ————→	–31172.57	(NPV after 5 cash flows)
9100 GSB 4 ————————→	–26971.76	(NPV after 6 cash flows)
2 GSB 3 9000 GSB 4 ————→	–20108.39	(NPV after 8 cash flows)
GSB 5 ————————————→	8.00	(checking that we've entered 8 periods cash flows so far)
4500 GSB 4 ————————→	–18696.99	(NPV after 9 cash flows)
71000 GSB 4 ———————→	879.93	(NPV after 10 cash flows)

Since the final NPV is positive, the investment exceeds the profit objectives.

Example 2:

The Cooper Company needs a new photocopier and is considering leasing the equipment as an alternative to buying. The end-of-the-year net cash cost of each option is:

PURCHASE

Year	Net Cash Cost
1	$ 533
2	948
3	1,375
4	1,815
5	2,270
Total Net Cash Cost	$6,941

LEASE

Year	Net Cash Cost
1	$1,310
2	1,310
3	1,310
4	1,310
5	1,310
Total Net Cash Cost	$6,550

Looking at total cost, leasing appears to be less. But, purchasing costs less the first two years. Mr. Cooper knows that he can make a 15% return on every dollar he puts in the business; the sooner he can reinvest money, the sooner he earns 15%. Therefore, he decides to consider the **timing of the costs,** discounting the cash flows at 15% to find the present value of the alternatives. Which option should he choose?

Keystrokes: **Outputs:**

PURCHASE

0 `GSB` `1` 15 `GSB` `2` 533 `GSB` `4`

948 `GSB` `4` 1375 `GSB` `4`

1815 `GSB` `4` 2270 `GSB` `4` ⟶   4250.71

LEASE

0 `GSB` `1` 5 `GSB` `3`

1310 `GSB` `4` ⟶————————⟶   4391.32

Leasing has a present value cost of $4391.32, while purchasing has a present value cost of $4250.71. Since these are both expense items, the lowest present value is the most desirable. So, in this case, purchase is the least costly alternative.

CALENDAR FUNCTIONS

For the period March 1, 1900 through February 28, 2100, this program solves for dates and days.

Given a date, the first part calculates an associated day number*. By using this program on two dates, the number of days between those dates may be found.

The second part takes a day number* and finds the corresponding date. The third part calculates the day of the week from a given day number*.

By using the first two parts together, a second date may be calculated from a date and a specified number of days (see example).

A date must be input in mm.ddyyyy format. For instance, June 3, 1975, is keyed in as 6.031975. It is important that the zero between the decimal point and the day of the month be included when the day of the month is less than 10. The day of the week is represented by the digits 0 through 6 where zero is Sunday.

Equations:

To calculate the day number from the date:

Julian Day number* = INT $(365.25\,y')$ + INT $(30.6001\,m')$ + d + 1,720,982

where:

$$y' = \begin{cases} \text{year} - 1 & \text{if } m = 1 \text{ or } 2 \\ \text{year} & \text{if } m > 2 \end{cases}$$

$$m' = \begin{cases} \text{month} + 13 & \text{if } m = 1 \text{ or } 2 \\ \text{month} + 1 & \text{if } m > 2 \end{cases}$$

Then days between dates is found by:

$$\text{Days} = \text{Day number}_2 - \text{Day number}_1$$

To calculate the date from a day number:

$$\text{Day \#} = \text{Julian Day Number*} - 1,720,982$$

$$y' = \text{INT}\left[\frac{\text{Day \#} - 122.1}{365.25}\right]$$

---

* The Julian Day number is an astronomical convention representing the number of days since January 1, 4713 B.C.

$$m' = INT \left[\frac{Day \ \# \ - \ INT \ (365.25 \ y')}{30.6001} \right]$$

$$Day \ of \ the \ month = Day \ \# - \ INT \ \left[365.25 \ y' \right]$$
$$- \ INT \ \left[30.6001 \ m' \right]$$

$$Month = m = \begin{cases} m' - 13 & \text{if } m' = 14 \text{ or } 15 \\ m' - 1 & \text{if } m' < 14 \end{cases}$$

$$Year = \begin{cases} y' & \text{if } m > 2 \\ y' + 1 & \text{if } m = 1 \text{ or } 2 \end{cases}$$

To calculate the day of the week:

$$Day \ of \ the \ week = 7 \times FRAC \ \left[(Day \ \# \ + \ 5)/7 \right]$$

Remarks:
- No checking is done to determine if input data represent valid dates.

01	*LBL1		50	ST09		
02	ENT↑	Break date input into the	51	RCL1		
03	INT	individual components of	52	x		
04	ST07	mm, dd, yyyy	53	INT		
05	-		54	ST-6		
06	EEX		55	RCL6	Calculate m'	
07	2		56	RCL2		
08	x		57	÷		
09	ENT↑		58	INT		
10	INT		59	ST07		
11	ST08		60	RCL6		
12	-		61	X⇄Y	Calculate day of month	
13	EEX		62	RCL2		
14	4		63	x		
15	x		64	INT		
16	ST09		65	-		
17	RCL7	m + 1	66	ST08		
18	1		67	RCL7		
19	+		68	1		
20	ENT↑		69	RCL8	Build (m'-1) .dd	
21	1/X	m + 1 → m'	70	%	Part of display	
22	.	y → y'	71	-		
23	7		72	-		
24	+		73	RCL7		
25	CHS		74	1		
26	INT		75	4		
27	ST+9		76	÷	Correct m'-1 and y' to m	
28	RCL4	If input to this routine has	77	INT	and y	
29	x	absolute value 1 or greater,	78	ST+9		
30	-	y = y ± 1	79	RCL4		
31	RCL2	m = m ± 2	80	x		
32	x		81	-		
33	INT		82	RCL9		
34	RCL9		83	EEX		
35	RCL1		84	6		
36	x		85	÷	Finish building	
37	INT		86	+	mm.ddyyyy result and	
38	+	Calculate day number	87	FIX6	display answer	
39	RCL8		88	RTN		
40	+		89	*LBL3		
41	FIX0		90	5		
42	RTN		91	+		
43	*LBL2		92	7		
44	ST06		93	÷	Calculate day of the week	
45	RCL3	Calculate y'	94	FRC	from day #	
46	-		95	7		
47	RCL1		96	x		
48	÷		97	RTN		
49	INT					

REGISTERS

0	1 365.25	2 30.6001	3 122.1	4 12	5
6 Day #	7 m	8 d	9 y	.0	.1 Used
.2 Used	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input constants for calculations:	365.25	**STO** 1	365.25
		30.6001	**STO** 2	30.6001
		122.1	**STO** 3	122.1
		12	**STO** 4	12
3	For day #, go to step 4. For			
	dates from day #, go to step 7.			
	For day of the week go to step 9.			
4	Input date and calculate day #	date	**GSB** 1	day #
5	Repeat step 4 for any other date			
6	For # of days between dates			
	calculate day #'s for each and			
	find the difference.	date 1	**GSB** 1	day #1
			STO • 1	day #1
		date 2	**GSB** 1	day #2
			RCL • 1	day #1
			−	difference
7	For dates from day #'s, input			
	day # and calculate date	day #	**GSB** 2	date
8	Repeat step 7 for any other			
	day #			
9	For day of the week from day #,			
	input day # and calculate day			
	of the week.	day #	**GSB** 3	0 ,..., 6
10	Repeat step 9 for any other			
	day #.			
11	For a new case, go to step 3.			

Example 1:

Senior Lieutenant Yuri Gagarin flew Vostok I into space on April 12, 1961. On July 21, 1969, Neil Armstrong set foot on the moon. How many days had passed between the first manned space flight and the moon landing? On what day of the week did each event take place.

Keystrokes:		**Outputs:**	
(Key in the program and store constants by:			
365.25 STO 1 30.6001 STO 2			
122.1 STO 3 12 STO 4)			
4.121961 GSB 1 STO · 1 →		716420.	(day # 1)
7.211969 GSB 1 STO · 2 →		719442.	(day # 2)
RCL · 1 − ⟶		3022.	(days)
RCL · 1 GSB 3 ⟶		3.	(Wednesday)
RCL · 2 GSB 3 ⟶		1.	(Monday)

Example 2:

A short term note is due in 200 days. If the issue date is June 11, 1976, what is the maturity date?*

Keystrokes:	**Outputs:**	
6.111976 GSB 1 ⟶	721959.	
200 + ⟶	722159.	
GSB 2 ⟶	12.281976	(December 28, 1976)

---

* First a day number is calculated for the known date, the number of days (200) is added to it, and this new day number is converted to a date.

Some securities use a 30/360 day calendar while this program performs all calculations using the actual number of days. Do not use the program for financial purposes unless you are sure that actual calendar days are correct.

MOON ROCKET LANDER

Imagine for a moment the difficulties involved in landing a rocket on the moon with a strictly limited fuel supply. You're coming down tail-first, freefalling toward a hard rock surface. You'll have to ignite your rockets to slow your descent; but if you burn too much too soon, you'll run out of fuel 100 feet up, and then you'll have nothing to look forward to but cold eternal moon rocks coming faster every second. The object, clearly, is to space your burns just right so that you will alight on the moon's surface with no downward velocity.

The game starts off with the rocket descending at a velocity of 50 feet/second from a height of 500 feet. The velocity and altitude are shown in a combined display as −50.0500, the altitude appearing to the right of the decimal point and the velocity to the left, with a negative sign on the velocity to indicate downward motion. Then the remaining fuel is displayed and the rocket fire count-down begins: "3", "2", "1", "0",. Exactly at zero you may key in a fuel burn. You only have one second, so be ready. A zero burn, which is very common, is accomplished by doing nothing. After a burn the sequence is repeated unless:

1. You have successfully landed—flashing zeros.

2. You have smashed into the lunar surface—flashing crash velocity.

You must take care, however, not to burn more fuel than you have; for if you do, you will free-fall to your doom! The final velocity shown will be your impact velocity (generally rather high). You have 60 units of fuel initially.

Equations:

We don't want to get too specific, because that would spoil the fun of the game; but rest assured that the program is solidly based on some old friends from Newtonian physics:

$$x = x_0 + V_0 t + \frac{1}{2} at^2, \quad V = V_0 + at, \quad V^2 = V_0^2 + 2a(x - x_0)$$

where:

x, V, a, and t are distance, velocity, acceleration, and time.

Remarks:

*   Only integer values for fuel burn are allowed. [R/S] can be used to stop Moon Rocket Lander at any time.

01	*LBL1			
02	5			
03	0	Store initial conditions		
04	0			
05	STO6			
06	5			
07	0			
08	CHS			
09	STO7			
10	6			
11	0			
12	STO8			
13	*LBL0			
14	RCL6			
15	FIX4	Divide height by 10000		
16	EEX	for proper display		
17	4			
18	÷			
19	RCL7	Build VV.Ohhh display,		
20	ABS	taking negative values into		
21	+	account		
22	RCL7			
23	X>0?			
24	GSB4			
25	X≷Y			
26	CHS			
27	PSE	Display VV.Ohhh		
28	PSE			
29	FIX0			
30	RCL8			
31	PSE	Count down for burn		
32	3			
33	PSE			
34	2			
35	PSE			
36	1			
37	PSE			
38	0			
39	PSE			
40	*LBL9	Accept input		
41	RCL8			
42	X≷Y	If fuel is gone calculate		
43	X>Y?	crash velocity		
44	GTO6			
45	ST-8			
46	2	Determine velocity and		
47	x	height		
48	5			
49	-			

50	STO9			
51	2			
52	÷			
53	RCL6			
54	+			
55	RCL7			
56	+			
57	RCL9			
58	ST+7			
59	R↓			
60	STO6			
61	INT			
62	X>0?	If no impact go for		
63	GTO0	another burn		
64	RCL7			
65	*LBL7	Flash crash velocity		
66	PSE			
67	GTO7			
68	*LBL6	Fuel exhausted:		
69	RCL8	call free-fall crash		
70	2	velocity		
71	.			
72	5			
73	-			
74	ST+6			
75	2			
76	x			
77	ST+7			
78	RCL6			
79	1			
80	0			
81	x			
82	RCL7			
83	X²			
84	+			
85	√X			
86	CHS			
87	GTO7			
88	*LBL4			
89	X≷Y			
90	CHS			
91	X≷Y			
92	RTN			

REGISTERS

0	1	2	3	4	5
6 X	7 V	8 Fuel	9 Accel.	.0	.1
.2	3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Assume manual control.		GSB 1	"V.ALT"
				"FUEL"
				"3"
				"2"
				"1"
				"0"
3	Key in burn upon "0" display:			
	Press and hold R/S until			
	blinking stops.		R/S	
	Enter burn	BURN	R/S	"V. ALT"
				"FUEL"
				"3"
				"2"
				"1"
				"0"
4	Go to step 3 until you land (flashing zeros) or crash (flashing impact velocity).			
5	If you survived last landing attempt, go to step 2 for another try.			

QUEEN BOARD

This game is based on the moves of a chess queen. A queen will be allowed to move only to the left, down, or diagonally to the left. The object of the game is to be the first player to move the queen to the lower left-hand corner of the chess board (square 158), by alternating moves between you and the calculator. You start by placing the queen on any square on the top row or right-hand column. This is your first move. The play then alternates.

The playing board is numbered as follows:

Start Zone
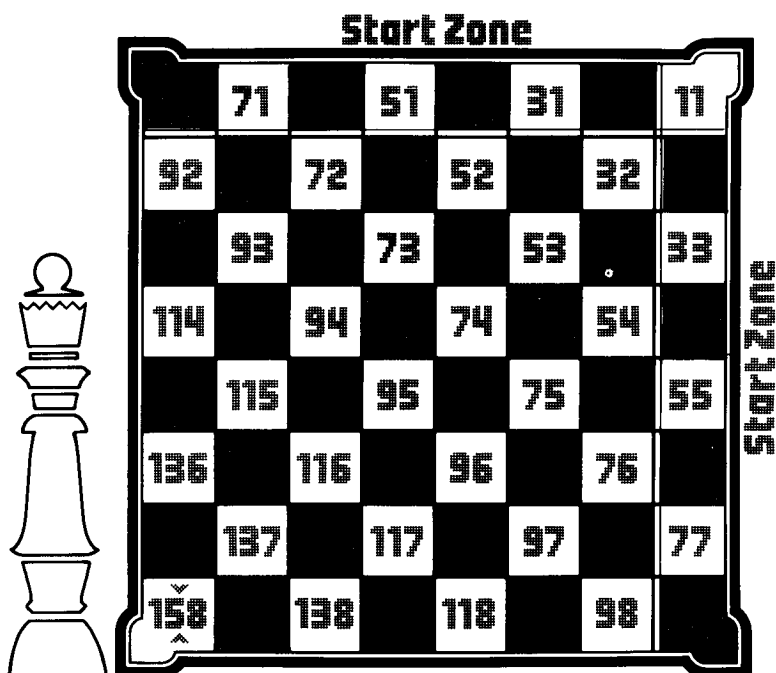
You tell the calculator your moves by keying in the number of the square you start on or move to. Press `GSB` `1` and the calculator responds with the square it moves to. Square 158 is the winning square.

The program does not check for illegal moves. If you win (by moving to square 158), the program will respond with 168 (the calculator acknowledges the loss by displaying a nonexistent square).

The program is in FIX 0 mode, for integer display.

Reference:

This program is based on an HP-65 Users' Library program by Jacob R. Jacobs.

Some interesting comments on the theory of "Queen Board" may be found in: Gardner, M. "Mathematical Games", Scientific American, vol 236, no 3., p. 134, March 1977.

01	*LBL1		50	RTN	
02	FIX0	Current position R_1	51	*LBL0	
03	ST01		52	1	
04	GSB0		53	5	
05	1		54	8	158 = R_2 ?
06	X=Y?		55	X=Y?	
07	GT08	$7 \to R_0$	56	GT06	
08	7		57	3	
09	ST00		58	1	127 = R_2 ?
10	*LBL9		59	-	
11	RCL1		60	X=Y?	
12	RCL0		61	GT06	
13	EEX		62	1	
14	1		63	-	126 = R_2 ?
15	x		64	X=Y?	
16	+	$10K + R_1 \to R_2$	65	GT06	
17	ST02	Position good?	66	5	
18	GSB0		67	1	
19	1		68	-	75 = R_2 ?
20	X=Y?	Yes, recall R_2	69	X=Y?	
21	GT07		70	GT06	
22	RCL0	$K+R_2 \to R_2$	71	2	
23	ST+2	Position good?	72	-	73 = R_2 ?
24	RCL2		73	X=Y?	
25	GSB0	Yes, recall R_2	74	GT06	
26	1		75	2	
27	X=Y?		76	9	
28	GT07		77	-	44 = R_2 ?
29	RCL0		78	X=Y?	
30	EEX		79	GT09	
31	1		80	3	
32	x	$10K + R_2 \to R_2$	81	-	41 = R_2 ?
33	ST+2		82	X=Y?	
34	RCL2	Position good?	83	GT06	
35	GSB0		84	RTN	
36	1		85	*LBL6	
37	X=Y?	Yes, recall R_2	86	1	
38	GT07		87	RTN	
39	DSZ				
40	GT09				
41	RCL1				
42	*LBL8				
43	EEX	Default move			
44	1	$10 + R_1 \to R_1$			
45	ST+1				
46	RCL1				
47	RTN				
48	*LBL7				
		Test for good position			

REGISTERS					
0 Indirect	1 Used	2 Used	3	4	5
6	7	8	9	0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Key in your starting position (first move).	Move	GSB 1	Calc's Move
3	Repeat step 2 until someone wins.			
	Display of 158: calculator wins			
	Display of 168: you win			
4	To begin new game, repeat step 2 with new starting position.			

Example:

Keystrokes: **Outputs:**

55 GSB 1 ⟶   75.

(You start on 55, and the calculator, after deep and careful thought, moves to 75).

97 GSB 1 ⟶   127.

(You respond with 97, and the calculator, showing no mercy moves to 127).

148 GSB 1 ⟶   158.

(You try 148, hoping the calculator's batteries run down before it can respond, but no luck—it wins by moving to 158).

BIORHYTHMS

From ancient days philosophers and sages have taught that human happiness lies in the harmonious integration of body, mind, and heart. Now a twentieth-century theory claims to be able to quantitatively gauge the functioning of these three aspects of our selves: the physical, sensitive, and cognitive.

The biorhythm theory is based on the assumption that the human body has inner clocks or metabolic rhythms with constant cycle times. Currently, three cycles starting at birth in a positive direction are postulated. The 23-day or physical cycle relates with physical vitality, endurance and energy. The 28-day cycle or sensitivity cycle relates with sensitivity, intuition and cheerfulness. The 33-day or cognitive cycle relates with mental alertness and judgement.

For each cycle, a day is considered either high, low, or critical. x is the output value for a given cycle. The high ($0 < x \leqslant 1$) times are regarded as energetic times, you are your most dynamic in the cycle. The low ($-1 \leqslant x < 0$) times are regarded as the recuperative periods. The critical days ($x = 0$) are regarded as your accident prone days, especially for the physical and sensitivity cycles.
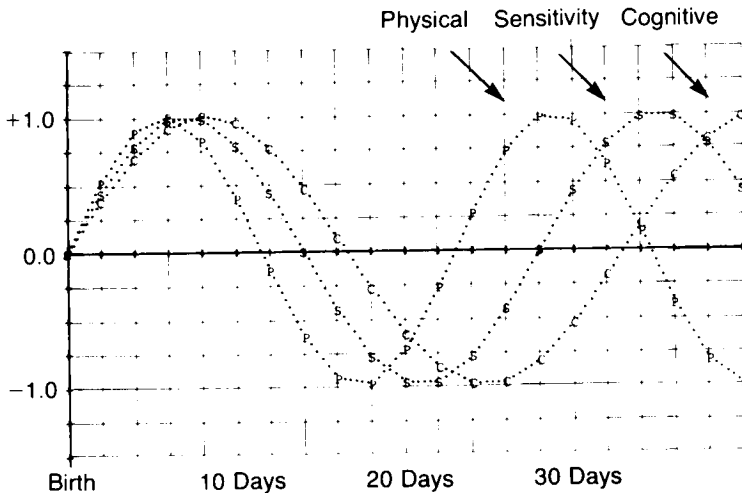
Remarks:

- The birthdate and biodate must be between January 1, 1901, and December 31, 2099.

- The format for input of dates is MM.DDYYYY. For example, June 3, 1976, is keyed in as 6.031976. The program does not check input data. Thus, if an improper format or an invalid date (e.g., February 30) is keyed in, erroneous answers may result.

- This program sets the angular mode to radians (RAD).

Steps	Notes	Steps	Notes
01 *LBL1		50 STO6	M
02 RAD	Birthdate store	51 –	
03 GSB8		52 EEX	
04 STO9	N₁	53 2	
05 RTN		54 x	
06 GSB0	Biodate	55 ENT↑	D
07 RCL9		56 INT	
08 –		57 STO5	
09 STO8	Store N₂ – N₁	58 –	
10 *LBL9		59 EEX	
11 1		60 4	
12 8		61 x	Y
13 STO7		62 STO4	
14 GSB8	23 Day cycle	63 2	
15 GSB8	28 Day cycle	64 RCL6	
16 *LBL8		65 X>Y?	
17 5	# Days	66 GTO6	
18 ST+7		67 1	
19 RCL8		68 ST–4	
20 RCL7		69 1	
21 ÷		70 2	
22 FRC		71 ST+6	
23 2		72 *LBL6	
24 x		73 1	
25 Pi		74 ST+6	
26 x		75 RCL6	
27 SIN		76 3	
28 ENT↑		77 0	
29 ABS		78 .	
30 X≠0?		79 6	
31 ÷		80 x	
32 LSTX		81 INT	
33 EEX		82 RCL4	
34 7		83 3	
35 +		84 6	
36 EEX		85 5	
37 7		86 .	
38 –		87 2	
39 x	***	88 5	
40 R/S	Bio value	89 x	
41 RTN		90 INT	
42 *LBL7		91 +	
43 1	Next day	92 RCL5	
44 ST+8		93 +	
45 GSB9		94 RTN	N
46 GTO7			
47 *LBL0	Compute N(M, D, Y,)		
48 ENT↑			
49 INT			

REGISTERS

0	1	2	3	4 Y	5 D
6 M	7 23,28,33	8 N₂ – N₁	9 N₁	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Key in birthdate	MM.DDYYYY	GSB 1	Day #*
3	Key in biodate and find bio values	MM.DDYYYY	R/S	P
			R/S	S
			R/S	C
4	To find bio values for succeed- ing days.		R/S R/S	P
			R/S	S
			R/S	C
5	For a new birthdate, go to step 2; for a new biodate, go to step 3.			
	* See Calendar Functions for explanation of this number.			

Example:

Calculate the bio values for June 29, 1976, for a person born March 27, 1948. Find the values for the two days following also.

Keystrokes: **Outputs:**

3.271948 GSB 1 ─────────▶	711656	(day #)
6.291976 R/S ─────────▶	−1.00	(June 29) (P)
R/S ─────────▶	−0.62	(S)
R/S ─────────▶	−1.00	(C)
R/S R/S ─────────▶	−0.98	(June 30) (P)
R/S ─────────▶	−0.78	(S)
R/S ─────────▶	−0.97	(C)
R/S ─────────▶	−0.89	(July 1) (P)
R/S ─────────▶	−0.90	(S)
R/S ─────────▶	−0.91	(C)

COUNTDOWN TIMER

This program provides a countdown timer and a calibration routine for measuring elapsed time. When using this program, you should remember that clock circuits of HP calculators are designed for calculator use, not for accurate time keeping. Although the routine may be calibrated quite accurately, highly stable performance should not be expected due to variable conditions about the calculator.

Equations:

$$Ca_{new} = Ca_{old} \frac{HP\ time}{Actual\ Time}$$

01 *LBL1*			
02 FIX4	Store constant		
03 STO2	- - - - - - - - - - -		
04 *LBL9*			
05 0	Alarm		
06 R/S	- - - - - - - - - - -		
07 STO1			
08 →H			
09 RCL2			
10 x	Store time		
11 STO0			
12 RCL1	- - - - - - - - - - -		
13 R/S	•••		
14 *LBL8*			
15 DSZ			
16 GTO8	Loop on counter		
17 GTO9	Go to "alarm"		
18 *LBL2*	- - - - - - - - - - -		
19 →H			
20 X≠Y			
21 →H			
22 X≠Y			
23 -			
24 RCL1	Calibrate constant		
25 →H			
26 ÷			
27 1/X			
28 RCL2			
29 x			
30 R/S			
31 GTO1			

REGISTERS					
0 counter	1 time	2 Ca	3	4	5
6	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input timer constant (try 10000)	Ca	GSB 1	0.0000
3	Input desired time	t(H.MMSS)	R/S	t
4	Start timer		R/S	0.0000
5	Timer loops for time t. When			
	0.0000 is displayed, time has			
	elapsed. For a new time t, exe-			
	cute step 3 and 4. To calibrate,			
	proceed to step 6.			
6	Input ending time and starting			
	time to calculate new constant	te	ENTER↑	
		ts	GSB 2	Ca
	To proceed depress		R/S	
	Then go to step 3.			

Example:

Measure elapsed times of 35 seconds and 1 minute 8 seconds.

Keystrokes: **Outputs:**

10000 GSB 1 ──────────→   0.0000
0.0035 R/S ──────────→   0.0035
R/S ──────────────→   0.0000

Timer runs for approximately 32 seconds.

For the second desired time:

Keystrokes: **Outputs:**

0.0108 R/S ──────────→   0.0108
R/S ──────────────→   0.0000

Supposing you had noticed the *actual* ending and starting times of the 2nd example were 9:58:03 and 9:57:01, respectively, then calibrate the timer with this information:

Keystrokes: **Outputs:**

9.5803 `ENTER ♦` 9.5701

`GSB` `2` ———————————→   10967.7421

`R/S` ———————————————→   0.0000

Now try the calibrated timer for 2 minutes 5 seconds:

0.0205 `R/S` ———————————→   0.0205

`R/S` ———————————————→   0.0000

Under the same conditions, the new timer constant 10967.7421 should be used for subsequent use of this program. Your HP calculator will have its own "best" constant for calibration.

BODY SURFACE AREA CALCULATIONS

This program calculates body surface area by either the Dubois or Boyd formula, ... allowing your choice of the preferred method. If cardiac output is known, cardiac index may also be calculated.

The Dubois is undefined, and should not be used, for children with a BSA of less than $0.6m^2$. If the result is less than 0.6, use the Boyd formula instead.

Data inputs are patient's height and weight, in either metric or English units, and if desired, the cardiac output. If the measurements are in English units (inches and pounds) the data are input as negative values and the program automatically converts then to metric units (cm and kilograms).

Equations:

Dubois formula:

$$BSA \ (m^2) = Ht \ (cm)^{0.725} \cdot Wt \ (kg)^{0.425} \cdot 71.84 \cdot 10^{-4}$$

Boyd formula:

$$BSA \ (m^2) = 3.207 \cdot Wt \ (gm)^{(0.7285 \ - \ 0.0188 \ \log \ Wt)} \cdot Ht \ (cm)^{0.3} \cdot 10^{-4}$$

Cardiac Index (CI):

$$CI = CO \ (\ell/min)/BSA \ (m^2)$$

Remarks:

- The height and weight may be input in either metric or English units. If English units are used, they must be entered as negative values, by pressing **CHS** after the number is input. Press **GSB** **1** to calculate BSA by the Dubois method, or **GSB** **2** for the Boyd result. The data must be reentered for calculation by the alternate method, if desired.

- Values for BSA calculated by the Dubois method are stored in Register 1 or, if by the Boyd method in Register 2 and may be recalled as needed.

- To calculate cardiac index: select BSA as calculated by the desired method and recall it from storage, then enter cardiac output and press **GSB** **3** .

01	*LBL1		50	3	
02	GSB0		51	1	
03	RCL8	Calculate BSA by DuBois	52	1	
04	.	method	53	8	
05	7		54	÷	
06	2		55	ST02	
07	5		56	RTN	•••
08	Yˣ		57	*LBL3	Input CO and calculate CI
09	RCL9		58	X⇄Y	
10	.		59	÷	
11	4		60	RTN	•••
12	2		61	*LBL0	
13	5		62	X<0?	Is ht. and wt. input
14	Yˣ		63	GSB9	metric or English?
15	x		64	ST09	
16	1		65	R↓	
17	3		66	X<0?	
18	9		67	GSB8	
19	.		68	ST08	
20	2		69	RTN	
21	÷		70	*LBL9	Convert wt. to metric
22	ST01		71	CHS	
23	RTN		72	2	
24	*LBL2	Calculate BSA by Boyd	73	.	
25	GSB0	method	74	2	
26	RCL8		75	÷	
27	.		76	RTN	
28	3		77	*LBL8	Convert ht. to metric
29	Yˣ		78	CHS	
30	RCL9		79	2	
31	EEX		80	.	
32	3		81	5	
33	x		82	4	
34	ENT↑		83	x	
35	LOG		84	RTN	
36	.				
37	0				
38	1				
39	8				
40	8				
41	x				
42	.				
43	7				
44	2				
45	8				
46	5				
47	-				
48	Yˣ				
49	÷				

REGISTERS

0	1 BSA (DuBois)	2 BSA (Boyd)	3	4	5
6	7	8 Ht.	9 Wt.	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input patient height (+ cm or –in)	Ht	ENTER↑	
3	Input patient weight (+ kg or –lb)	Wt		
4	Calculate BSA			
	by Dubois formula		GSB 1	BSA (m²)
	or, by Boyd formula		GSB 2	BSA (m²)
	Note: Reenter data before per-			
	forming calculation again by			
	alternate method.			
5	Calculate cardiac index. Recall			
	desired BSA from storage	BSA, Dubois	RCL 1	BSA (m²)
	or,	BSA, Boyd	RCL 2	BSA (m²)
	Input cardiac output and	CO, ℓ/min	GSB 3	CI,(ℓ/min/m²)
	calculate cardiac index			

Example:

A patient has the following height and weight.

Ht = 60 in or 152.40 cm
Wt = 100 lbs or 45.45 kg.

Calculate BSA by both the Dubois and Boyd methods. If the cardiac output, (CO) is 5 1/min calculate the cardiac index using the Dubois BSA.

Keystrokes: **Outputs:**

60 CHS ENTER↑
100 CHS GSB 1 ————————→   1.39   m² (Dubois)
152.4 ENTER↑ 45.45 GSB 2 —→   1.40   m² (Boyd)
RCL 1 5 GSB 3 ————————→   3.59   CI (by Dubois)

PULMONARY FUNCTIONS AND VITAL CAPACITY

The pulmonary function testing package provides calculations of the predicted and percent predicted values for vital capacity (VC), forced expiratory volume after 1 second (FEV_1), maximum expiratory flow rate (MEFR), maximum ventilatory volume after 12 seconds (MVV_{12}), residual volume (RV), total lung capacity (TLC), functional residual capacity (FRC), and forced expiratory flow from 25% to 75% (FEF 25%-75%).

The calculations are performed for either male or female patients, given the patient's height and age.

Equations:

All of the functions (with two exceptions) are calculated from a general equation of the form: $(A \cdot Ht(cm)) - (B \cdot AGE(years)) - C$, where A, B, and C are constants given in Table 1.

The exceptions are:

- Female TLC: If height is greater than 174 cm (68.5 inches) add 1 cm to height before calculation.

- Female Predicted FEF: $(A \cdot Ht(cm)) - (B \cdot AGE(years)) - (0.00005 \cdot AGE^2(years)) - C$.

$$25\% \ VC = 0.25 \ VC$$
$$75\% \ VC = 0.75 \ VC$$
$$\Delta t = t_{75\%} - t_{25\%}$$
$$\text{Measured FEF} = (0.5 \cdot VC)/\Delta t$$

References:

Morris, J.F., Koski, A., Johnson, L.C., American Rev. Resp. Dis., 1971, *103*, 57.

Bates, et. al., Respiratory Function in Disease, W.G. Saunders Co., 1971.

Table 1
Constants For Calculation of Predicted Values

	FEMALE			MALE		
	A	**B**	**C**	**A**	**B**	**C**
Predicted VC	0.045	0.024	2.852	0.058	0.025	4.24
Predicted FEV$_1$	0.035	0.025	1.932	0.036	0.032	1.26
Predicted MEFR	0.057	0.036	2.532	0.043	0.047	-2.07
Predicted MVV$_{12}$	0.762	0.81	6.29	0.9	1.51	-27.0
Predicted RV	0.024	-0.012	2.63	0.03	-0.015	3.75
Predicted TLC*	0.078	0.01	7.36	0.094	0.015	9.17
Predicted FRC	0.047	0.00	4.86	0.051	0.00	5.05
Predicted FEF	0.02	0.03	-1.3	0.02	0.04	-2.0

Detailed User Instructions:

Key in the program. Then key in the patient height, in centimeters or inches (if in inches, input as a negative number) and press **GSB** ①. Then key in patient age in years and press **GSB** ②. Now any of the predicted values may be calculated by entering the appropriate constants A, B and C from table 1 and pressing **GSB** ③. The predicted value of the function is displayed. Key in the measured value of the function and press **R/S** to obtain the percent of predicted value.

The measured forced expiratory flow rate from the 25% and 75% points of a spirogram and predicted and percentage of predicted value are calculated as follows:

Enter A, B, and C from table 1, then press **GSB** ④. The predicted FEF is displayed. Key in the vital capacity as measured from the spirogram and press **R/S**. The display will show 25% VC. Read the measured time of this volume from the spirogram, key in this time in seconds and press **R/S**. The display will now show 75% VC. Determine the time at this volume from the spirogram, key it in and again press **R/S**. The measured FEF is now displayed. Pressing **R/S** again results in display of the percentage of predicted value. If desired, the predicted FEF can be recalled by pressing **R/S** or **RCL** ⓪.

*(Note: for female patients over 174 cm in height be sure to add 1 cm to height before calculating TLC, then reenter proper value for height before proceeding with calculations of other functions).

01	*LBL1*			50	7		
02	X<0?			51	5		
03	*GSB0*	Store ht.		52	x	Input Δt @ .75 VC	
04	*ST04*			53	RTN	Calculate measured FEF	
05	RTN			54	X≷Y		
06	*LBL2*			55	R↓		
07	*ST05*	Store age		56	X≷Y		
08	RTN			57	–		
09	*LBL3*			58	.		
10	*ST03*			59	5		
11	R↓	Calculate functions		60	RCL6		
12	*ST02*			61	x		
13	R↓			62	X≷Y		
14	*ST01*			63	÷	*** Display Meas. FEF	
15	RCL1			64	R/S		
16	RCL4			65	RCL0		
17	x			66	÷		
18	RCL2			67	EEX		
19	RCL5			68	2		
20	x			69	x	*** Display % of pre-	
21	–			70	R/S	dicted FEF.	
22	RCL3			71	RCL0		
23	1			72	RTN		
24	.			73	*LBL0*		
25	3			74	CHS	Change inches to cm	
26	CHS			75	2		
27	X=Y?			76	.		
28	GSB9			77	5		
29	R↓			78	4		
30	–			79	x		
31	*ST00*			80	RTN		
32	RTN	*** Display function		81	*LBL9*		
33	X≷Y			82	R↓		
34	÷	Input meas. value		83	5	Calculate female FEF.	
35	EEX			84	EEX		
36	2			85	5		
37	x	Calculate and display % of		86	CHS		
38	RTN	predicted		87	RCL5		
39	*LBL4*			88	X²		
40	GSB3	Calculate predicted FEF		89	x		
41	R/S			90	+		
42	*ST06*			91	ENT↑		
43	.			92	RTN		
44	2						
45	5						
46	x						
47	RTN	Input Δt @ .25VC					
48	RCL6						
49	.						

REGISTERS

0 Predicted Value	1 A	2 B	3 C	4 ht. (cm)	5 age (Years)
6 Measured VC	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input patient height in cm, or	Ht, cm.	`GSB` `1`	Ht, cm
	in inches	Ht, in.	`CHS` `GSB`	Ht, cm
			`1`	
3	Input patient age in years	AGE, Yrs.	`GSB` `2`	AGE, Yrs.
4	Calculate predicted values of			
	desired functions.			
	Input A from table I	A	`ENTER↑`	
	Input B from table I	B	`ENTER↑`	
	Input C from table I	C	`GSB` `3`	Pred. Value
5	Calculate % of predicted value			
	Input measured value	Meas. Value	`R/S`	% of Pred.
6	Calculate forced expiratory flow			
	Calculate predicted FEF			
	Input A from table I	A	`ENTER↑`	
	Input B from table I	B	`ENTER↑`	
	Input C from table I	C	`GSB` `4`	FEF Pred.
	Input measured VC	VC	`R/S`	25% VC
	Obtain t @ 25% VC from			
	spirogram and input	$t_{25\%}$ sec.	`R/S`	75% VC
	Obtain t @ 75% VC from			
	spirogram and input	$t_{75\%}$ sec.	`R/S`	$FEF_{Meas.}$
	Calculate % predicted FEF		`R/S`	% $FEF_{Pred.}$
	Recall $FEF_{Pred.}$ if desired		`R/S`	$FEF_{Pred.}$

Example 1:

Calculate the predicted and percentage of predicted vital capacity, residual volume and forced expiratory flow for a male 6 feet tall, 28 years of age.

Measured values are:

VC = 5.2 ℓ
RV = 2.0 ℓ

Keystrokes: **Outputs:**

Calculate VC: From table 1, A = 0.058, B = 0.025, C = 4.24

72 CHS GSB 1 ⟶   182.88   (cm)
28 GSB 2 ⟶   28   (years)
.058 ENTER↑ .025 ENTER↑ 4.24
GSB 3 ⟶   5.67   (ℓ, VC$_{Pred.}$)
5.2 R/S ⟶   91.76   (% Pred.)

Calculate RV: From table 1, A = 0.03, B = –0.015, C = 3.75

.03 ENTER↑ .015 CHS ENTER↑
3.75 GSB 3 ⟶   2.16   (ℓ, RV$_{Pred.}$)
2 R/S ⟶   92.75   (% Pred.)

Calculate % of FEF: From table 1, A = 0.02, B = 0.04, C = –2.0

.02 ENTER↑ .04 ENTER↑ 2.0
CHS GSB 4 ⟶   4.54   (ℓ, FEF$_{Pred.}$)

Input Measured VC = 5.2 ℓ.

5.2 R/S ⟶   1.30   (25% VC)

From Spirogram at 25% VC = 1.3 Obtain $t_{25\%}$ = 0.4 sec.

.4 R/S ⟶   3.90   (75% VC)

From Spirogram at 75% VC = 3.9 Obtain $t_{75\%}$ = 1.0 sec.

1 R/S ⟶   4.33   (ℓ, FEF)
R/S ⟶   95.50   (% Pred.)
R/S ⟶   4.54   (ℓ, FEF$_{Pred.}$)

Example 2:

Calculate the predicted and percentage of predicted vital capacity for a female
patient 5 feet tall, 28 years of age.

Measured VC = 3.0 ℓ
Measured RV = 1.2 ℓ

Keystrokes: **Outputs:**

Calculate VC: From table 1, A = 0.045, B = 0.024, C = 2.852

60 CHS GSB 1 ⟶   152.40   (cm)
28 GSB 2 ⟶   28.00   (years)
.045 ENTER↑ .024 ENTER↑
2.852 GSB 3 ⟶   3.33   (ℓ, VC$_{Pred.}$)
3.0 R/S ⟶   89.98   (% Pred.)

Calculate R V: From table 1, A = 0.024, B = –0.012, C = 2.63

.024 **ENTER♦** .012 **CHS** **ENTER♦**

2.63 **GSB** **3** ——————————————→   1.36   (ℓ, RV$_{Pred.}$)

1.2 **R/S** ——————————————→   88.00   (% Pred.)

Calculate % FEF: From table 1, A = 0.02, B = 0.03, C = –1.3

.02 **ENTER♦** .03 **ENTER♦** 1.3

CHS **GSB** **4** ——————————————→   3.47   (ℓ, FEF$_{Pred.}$)

3.0 **R/S** ——————————————→   0.75   (25% VC)

From Spirogram Find t = 0.4 sec @ 25% VC

.4 **R/S** ——————————————→   2.25   (75% VC)

From Spirogram Find t = 1.0 sec @ 75% VC

1.0 **R/S** ——————————————→   2.50   (ℓ, FEF$_{Meas.}$)

R/S ——————————————→   72.07   (% Pred.)

R/S ——————————————→   3.47   (ℓ, FEF$_{Pred.}$)

GREAT CIRCLE NAVIGATION

This program calculates the great circle distance between two points and the initial course from the first point. Coordinates are input in degrees-minutes-seconds format. The distance is displayed in nautical miles and the initial course in decimal degrees.
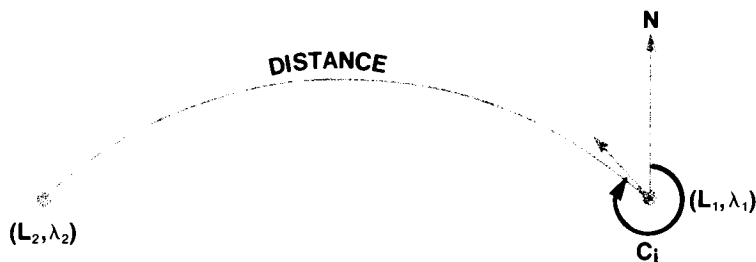
Equations:

$$D = 60 \cos^{-1}\left[\sin L_1 \sin L_2 + \cos L_1 \cos L_2 \cos(\lambda_2 - \lambda_1)\right]$$

$$C = \cos^{-1}\left[\frac{\sin L_2 - \sin L_1 \cos(D/60)}{\sin(D/60)\cos L_1}\right]$$

$$C_i = \begin{cases} C; \sin(\lambda_2 - \lambda_1) < 0 \\ 360 - C; \sin(\lambda_2 - \lambda_1) \geqslant 0 \end{cases}$$

where:

L_1, λ_1 = coordinates of initial point

L_2, λ_2 = coordinates of final point

D = distance from initial to final point

C_i = initial course from initial to final point

Remarks:

Southern latitudes and eastern longitudes must be entered as negative numbers.

Truncation and round off errors occur when the source and destination are very close together (1 mile or less).

Do not use coordinates located at diametrically opposite sides of the earth.

Do not use latitudes of +90° or −90°.

Do not try to compute initial heading along a line of longitude ($L_1 = L_2$).

This program assumes the calculator is set in DEG mode.

01 *LBL0		50 SIN	
02 →H		51 ÷	
03 STO0	L₁	52 COS⁻¹	C
04 RTN	– – – – – – – – –	53 RCL4	
05 →H		54 SIN	
06 STO1	λ₁	55 X<0?	
07 R/S	– – – – – – – – –	56 GTO9	
08 →H		57 R↓	
09 STO2	L₂	58 3	
10 R/S	– – – – – – – – –	59 6	
11 →H		60 0	
12 STO3	λ₂	61 X⇄Y	
13 R/S	– – – – – – – – –	62 –	
14 *LBL1		63 RTN	*** Cᵢ
15 RCL0		64 *LBL9	
16 SIN		65 R↓	
17 RCL2		66 RTN	*** Cᵢ
18 SIN			
19 x			
20 RCL0			
21 COS			
22 RCL2			
23 COS			
24 x			
25 RCL3			
26 RCL1			
27 –			
28 STO4			
29 COS			
30 x			
31 +			
32 STO5			
33 COS⁻¹			
34 STO6			
35 6			
36 0			
37 x			
38 R/S	*** D		
39 RCL2			
40 SIN			
41 RCL0			
42 SIN			
43 RCL5			
44 x			
45 –			
46 RCL0			
47 COS			
48 ÷			
49 RCL6			

REGISTERS					
0 L₁	1 λ₁	2 L₂	3 λ₂	4 λ₂ – λ₁	5 COS D/60
6 D/60	7 Cᵢ	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Key in latitude and longitude of origin.	L_1 (D.MS)	GSB 0	L_1 (dec. deg)
		λ_1 (D.MS)	R/S	λ_1 (dec. deg)
3	Key in latitude and longitude of destination.	L_2 (D.MS)	R/S	L_2 (dec. deg)
		λ_2 (D.MS)	R/S	λ_2 (dec. deg)
4	Calculate distance and initial course.		GSB 1	D (n.m.)
			R/S	C_i (dec. deg)

Example 1:

Find the distance and initial course for the great circle from Tokyo (L35°40′N, λ139°45′E) to San Francisco (L37°49′N, λ122°25′E).

Keystrokes: **Outputs:**

35.40 GSB 0 139.45

CHS R/S ⟶   −139.75

37.49 R/S 122.25 R/S ⟶   122.42

GSB 1 ⟶   4460.04   (D, n. m.)

R/S ⟶   54.37   (C_i, dec. deg.)

Example 2:

What is the distance and initial great circle course from L33°53′30″S, λ18°23′10″E to L40°27′10″N, λ73°49′40″W?

Keystrokes: **Outputs:**

33.533 CHS GSB 0 18.231

CHS R/S ⟶   −18.39

40.271 R/S 73.494 R/S ⟶   73.83

GSB 1 ⟶   6763.09   (D, n. m.)

R/S ⟶   304.48   (C_i, dec. deg.)

RHUMB LINE NAVIGATION

This program is designed to assist in the activity of course planning. You supply the latitude and longitude of the point of origin and the destination. The program calculates the rhumb line course and the distance from origin to the destination.

Since the rhumb line is the constant course path between points on the globe, it forms the basis of short distance navigation. In low and midlatitudes the rhumb line is sufficient for virtually all course and distance calculations which navigators encounter. However, as distance increases or at high latitudes the rhumb line ceases to be an efficient track since it is not the shortest distance between points.

The shortest distance between points on a sphere is the great circle. However, in order to steam great circles, an infinite number of course changes are necessary. Since it is impossible to calculate an infinite number of courses at an infinite number of points, several rhumb lines may be used to approximate a great circle. The more rhumb lines used the closer to the great circle distance the sum of the rhumb line distances will be. The Great Circle Navigation program may be used to calculate intermediate course change points which can be linked by rhumb lines.
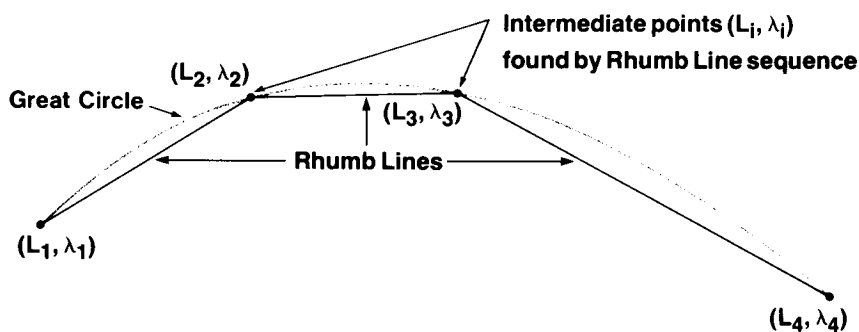
Latitudes and longitudes are input in degrees-minutes-seconds. Course is displayed in decimal degrees. Southern latitudes and eastern longitudes are input as negative numbers.

Equations:

$$C = \tan^{-1} \frac{\pi (\lambda_1 - \lambda_2)}{180 (\ln \tan (45 + \frac{1}{2} L_2) - \ln \tan (45 + \frac{1}{2} L_1))}$$

$$D = \begin{cases} 60 (\lambda_2 - \lambda_1) \cos L; \cos C = 0 \\ \\ 60 \dfrac{(L_2 - L_1)}{\cos C} \; ; \text{otherwise} \end{cases}$$

where:

(L_1, λ_1) = position of initial point

(L_2, λ_2) = position of final point

D = rhumb line distance

C = rhumb line course

Remarks:

No course should pass through either the south or north pole.

Errors in distance calculations may be encountered as cos C approaches zero.

Accuracy deteriorates for very short legs.

This program assumes the calculator is set in DEG mode.

01 *LBL1		50 RTN		
02 →H		51 *LBL8		
03 STO3	λ_2	52 3		
04 R↓		53 6	E to W 360 − C	
05 →H		54 0		
06 STO2	L_2	55 RCL5		
07 R↓		56 ABS		
08 →H	λ_1	57 −		
09 STO1		58 *LBL7		
10 R↓		59 ABS		
11 →H	L_1	60 STO6		
12 STO0		61 1		
13 FIX2	$\lambda_1 - \lambda_2$	62 8		
14 RCL1		63 0		
15 RCL3	_ _ _ _ _ _	64 RCL4		
16 −		65 ABS		
17 STO4		66 X≤Y?	is $[\lambda_1 - \lambda_2] > 180°$?	
18 2	Make − 180 ≤ $\lambda_1 - \lambda_2$	67 GSB6	If so subtract from 360	
19 ÷	≤ 180	68 RCL2		
20 SIN		69 COS		
21 SIN⁻¹		70 x		
22 9		71 STO7		
23 0		72 RCL2		
24 ÷		73 RCL0		
25 Pi		74 −		
26 x		75 RCL5		
27 RCL2		76 COS		
28 GSB9		77 X≠0?	is C = 90°?	
29 RCL0		78 ÷		
30 GSB9		79 ENT↑		
31 −		80 X=0?		
32 →P		81 RCL7		
33 R↓	C	82 6		
34 STO5		83 0		
35 RCL4		84 x		
36 SIN		85 ABS		
37 SIN⁻¹		86 R/S	*** Distance	
38 X<0?		87 RCL6		
39 GTO8	x < 0 means east to west,	88 RTN	*** Course	
40 RCL5		89 *LBL6	_ _ _ _ _ _ _ _	
41 GTO7		90 3		
42 *LBL9	If west to east	91 6	If $[\lambda_1 - \lambda_2] > 180°$	
43 2	C is answer	92 0		
44 ÷		93 X≷Y		
45 4		94 −	then 360 − $[\lambda_1 - \lambda_2]$	
46 5		95 RTN		
47 +				
48 TAN				
49 LN				

REGISTERS					
0 L_1	1 λ_1	2 L_2	3 λ_2	4 $\lambda_1 - \lambda_2$	5 Used
6 C	7 Used	8	9	.0	.1
.2	3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Key in latitude and longitude of origin			
		L₁ (D.MS)	ENTER↑	
		λ₁ (D.MS)	ENTER↑	
3	Key in latitude and longitude of destination			
		L₂ (D.MS)	ENTER↑	
		λ₂ (D.MS)		
4	Calculate distance and course		GSB 1	D (n.m.)
			R/S	C (dec. deg.)
	Note: Southern latitudes and eastern longitudes must be input as negative numbers.			

Example 1:

What is the distance and course from L35°24′12″N, λ125°02′36″W to L41°09′12″N, λ147°22′36″E?

Keystrokes: **Outputs:**

35.2412 ENTER↑ 125.0236
ENTER↑ 41.0912 ENTER↑
147.2236 CHS GSB 1 ⟶   4135.60   (DIST., n. m.)
R/S ⟶   274.79   (C, dec. deg.)

Example 2:

What course should be sailed to travel a rhumb line from L2°13′42″S, λ179°07′54″E to L5°27′24″N, λ179°24′36″W? What is the distance?

Keystrokes: **Outputs:**

2.1342 CHS ENTER↑ 179.0754
CHS ENTER↑ 5.2724 ENTER↑
179.2436 GSB 1 ⟶   469.31   (DIST., n. m.)
R/S ⟶   10.73   (C, dec. deg.)

SIGHT REDUCTION TABLE

This program calculates the computed altitude, Hc, and azimuth, Zn, of a celestial body given the observer's latitude, L, and the local hour angle, LHA, and declination, (d), of the body. It thus becomes a replacement for the nine volumes of H0 214. Moreover, the user need not bother with the distinctions of same name and contrary name; the program itself resolves all ambiguities of this type.

Equations:

$$Hc = \sin^{-1}\left[\sin d \sin L + \cos d \cos L \cos LHA\right]$$

$$Zn = \begin{cases} Z; & \sin LHA < 0 \\ 360 - Z; & \sin LHA \geq 0 \end{cases}$$

$$Z = \cos^{-1}\left[\frac{\sin d - \sin L \sin Hc}{\cos L \cos Hc}\right]$$

Remarks:

● Southern latitudes and southern declinations must be entered as negative numbers.

● The meridian angle t may be input in place of LHA, but if so, eastern meridian angles must be input as negative numbers.

● The program assumes the calculator is set in DEG mode.

Note:

This program may also be used for star identification by entering observed azimuth in place of local hour angle and observed altitude in place of declination. The outputs are then declination and local hour angle instead of altitude and azimuth. The star may be identified by comparing this computed declination to the list of stars in *The Nautical Almanac*.

01	***LBL1**		**50**	**3**	
02	**→H**		**51**	**6**	
03	**STO0**	L	**52**	**0**	
04	**RTN**	– – – – – – – – – –	**53**	**X⇄Y**	
05	**→H**		**54**	**–**	
06	**STO1**		**55**	**RTN**	*** Zn
07	**R/S**		**56**	***LBL0**	
08	**→H**		**57**	**R↓**	
09	**STO2**	d	**58**	**RTN**	*** Zn
10	**RCL0**	– – – – – – – – – –			
11	**SIN**	LHA			
12	**RCL1**				
13	**SIN**				
14	**×**				
15	**RCL0**				
16	**COS**				
17	**RCL1**				
18	**COS**				
19	**×**				
20	**RCL2**				
21	**COS**				
22	**×**				
23	**+**				
24	**STO3**				
25	**SIN⁻¹**				
26	**STO4**	Hc, dec. deg.			
27	**→HMS**				
28	**FIX4**	Hc, D.MS			
29	**R/S**	***			
30	**FIX1**				
31	**RCL1**				
32	**SIN**				
33	**RCL3**				
34	**RCL0**				
35	**SIN**				
36	**×**				
37	**–**				
38	**RCL0**				
39	**COS**				
40	**÷**				
41	**RCL4**				
42	**COS**				
43	**÷**				
44	**COS⁻¹**				
45	**RCL2**				
46	**SIN**	Z			
47	**X<0?**				
48	**GTO0**				
49	**R↓**				

REGISTERS					
0 L	1 d	2 LHA	3 Sin Hc	4 Hc	5
6	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTION	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input the following:			
	Observer's latitude	L (D.MS)	GSB 1	
	Declination	d (D.MS)	R/S	
	Local hour angle	L.H.A. (D.MS)		
3	Calculate:			
	Altitude		R/S	Hc (D.MS)
	Azimuth		R/S	Zn (dec. deg.)
	or			
2	Input:			
	Observer's latitude	L (D.MS)	GSB 1	
	Altitude	Hc (D.MS)	R/S	
	Azimuth	Zn (D.MS)		
3	Calculate:			
	Declination		R/S	d (D.MS)
	Local hour angle		R/S	L.H.A. (dec. deg.)

Example 1:

Calculate the altitude and azimuth of the moon if its LHA is 2°39′54″W and its declination 13°51′06″S. The assumed latitude is 33°20′N.

Keystrokes: **Outputs:**

33.20 GSB 1 ⟶   33.33
13.5106 CHS R/S ⟶   −13.85
2.3954 R/S ⟶   42.4447   (Hc, D.MS)
R/S ⟶   183.5   (Zn, dec. deg.)

Example 2:

Calculate the altitude and azimuth of REGULUS if its LHA is 36°39′18″W and its declination is 12°12′42″N. The assumed latitude is 33°30′N.

Keystrokes: **Outputs:**

33.30 GSB 1 ⟶   33.5
12.1242 R/S ⟶   12.2
36.3918 R/S ⟶   50.2425   (Hc, D.MS)
R/S ⟶   246.3   (Zn, dec. deg.)

Example 3:

At 6:10 G.M.T. on January 12, 1977 a star peeked through the clouds over Corvallis (L44°34′N, λ123°17′W). An alert observer using a bubble sextant quickly determined its altitude to be 26° and its azimuth 158°. Using *The Nautical Almanac* identify the star.

Keystrokes:	**Outputs:**	
44.34 **GSB** 1 26 **R/S**		
158 **R/S** ——————————→	−16.3725	(d, D.MS)
R/S ——————————————→	339.4	(L.H.A., dec. deg.)

Obtain G.H.A. by adding latitude to L.H.A.

123.17 **9** **→H** **+** ——————→	462.7	(G.H.A., dec. deg.)

Then convert G.H.A. to S.H.A. by subtracting G.H.A. ARIES (for 6:10 G.M.T., January 12, 1977 G.H.A. ARIES is 203.4 dec. degrees).

203.4 **−** **f** **→H.MS** ——————→	259.2	(S.H.A., D.MS)

From *The Nautical Almanac* we find the star to be SIRIUS (S.H.A. = 258°58.1′, d = S16°41.2′).

NEWTON'S METHOD-SOLUTION TO f(x) = 0

This program uses Newton's method to find a solution for $f(x) = 0$, where $f(x)$ is specified by the user.

The user must define the function $f(x)$ by keying into program memory the keystrokes required to find $f(x)$, assuming x is in the X-register. 55 program steps are available for defining $f(x)$; the program only uses registers R_0 through R_4, the rest of the registers are available to the user.

The user must provide the program with an initial guess, x_1, for the solution. The closer the initial guess is to the actual solution, the faster the program will converge to an answer. The program will halt when two successive approximations for x, say x_i and x_{i+1}, are within a tolerance ϵ, i.e., when $[x_{i+1} - x_i]$ $< \epsilon$. The value for ϵ must be input by the user. In general a reasonable value for ϵ might be $10^{-6} x_1$.

Equations:

The basic formula used by Newton's method to generate the next approximation for the solution is:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$
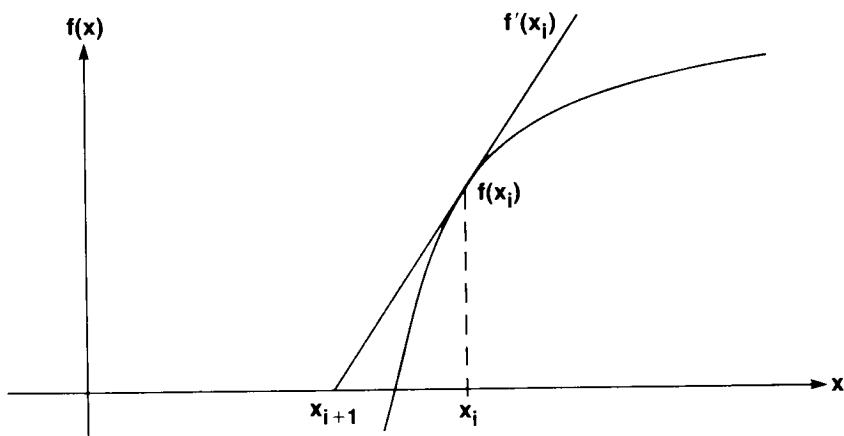
This program makes a numerical approximation for the derivative $f'(x)$ to give the following equation:

$$x_{i+1} = x_i - \delta_i \left[\frac{f(x_i + \delta_i)}{f(x_i)} - 1 \right]^{-1}$$

where:

$$\delta_i = 10^{-5} x_i$$

Remarks:

After the routine has finished calculating, the last value of f(x) may be displayed by pressing **RCL** **4** . If this value is not close enough to zero, the program may be run again with a smaller value for ϵ.

Programming Remarks:

This is one of the more complex programs in the book. The main difficulty is that at each iteration both f(x) and f(x + δ) need to be calculated, but the function f is keyed in in only one place in program memory. Large computers handle this problem by the use of a subroutine. This program simulates that technique by a number stored in R_0 known as a flag. The flag is set to 0 to indicate that f(x) is to be calculated, or to 1 if f(x + δ) is to be found. After the calculation of f, a test is made on the flag. If it is 0, the program will branch to an instruction which will store f(x); if it is 1, the program will go on to calculate a derivative based on f(x + δ).

01 *LBL1			
02 STO2	Store x, ε		
03 X⇄Y			
04 STO1	– – – – – – – – – – – –		
05 *LBL8			
06 CLX			
07 STO0	Set flag to 0 for f(x)		
08 RCL1			
09 GTO0	– – – – – – – – – – – –		
10 *LBL6			
11 R↓			
12 STO4			
13 1			
14 STO0			
15 RCL1	Store f(x) and calculate δ		
16 RCL1			
17 EEX			
18 5			
19 ÷			
20 STO3	– – – – – – – – – – – –		
21 +	User's f(x)		
22 *LBL0	– – – – – – – – – – – –		
23 *LBL7			
24 X=0?			
25 GTO9			
26 RCL0			
27 X=0?			
28 GTO6			
29 R↓			
30 RCL4	Calculate x_{i+1}		
31 ÷			
32 1	x_i and		
33 –			
34 1/X	$[x_{i+1} - x_i] > \epsilon$?		
35 RCL3			
36 x			
37 ST–1			
38 ABS			
39 RCL2			
40 X⇄Y			
41 X>Y?			
42 GTO8	– – – – – – – – – – – –		
43 *LBL9			
44 RCL1	•••		
45 RTN	Output x_0		

REGISTERS					
0 Flag	1 x	2 ε	3 δ	4 f(x)	5
6	7	8	9	0	.1
.2	.3	4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Press `GTO` `0`		`GTO` `0`	
3	Switch to PRGM and key			
	in function f(x)			
4	Switch to RUN			
5	Input initial guess for solution			
	and tolerance to calculate			
	solution.	x_1	`ENTER♦`	
		ϵ	`GSB` `1`	x_0
6	To recall the last f(x)		`RCL` `4`	f(x)

Example:

Find a root x_0 of the equation $\ln x + 3x - 10.8074 = 0$ in the interval $\left[1, 5\right]$. An accuracy of 10^{-4} is acceptable.

Keystrokes: **Outputs:**

`GTO` `0` ,

Switch to PRGM

`f` `ln` `f` `LAST X` 3 `×`
`+` 10.8074 `−`

Switch to RUN

1 `ENTER♦` `EEX` `CHS` 4
`GSB` `1` ⟶ 3.21 (root)
`RCL` `4` ⟶ −1.50   −07   (f(3.21))

NUMERICAL INTEGRATION *BY SIMPSON'S FORMULA*

This program will perform numerical integration by Simpson's formula whether a function is known explicitly or only at a finite number of equally spaced points (discrete case).

Discrete Case:

Let x_0, x_1,..., x_n be n equally spaced points ($x_j = x_0 + jh$, $j = 1, 2,..., n$) at which corresponding values $f(x_0)$, $f(x_1)$,..., $f(x_n)$ of the function $f(x)$ are known. The function itself need not be known explicitly. After input of the step size h and the values of $f(x_j)$, $j = 0, 1,..., n$, then the integral

$$\int_{x_0}^{x_n} f(x)\, dx \tag{1}$$

may be approximated using Simpson's rule:

$$\int_{x_0}^{x_n} f(x)\, dx \approx \frac{h}{3} \Big[\, f(x_0) + 4f(x_1) + 2f(x_2) + $$
$$\ldots + 4f(x_{n-3}) + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\Big] \tag{2}$$

In order to apply Simpson's rule, n must be even.

Explicit Functions:

If an explicit formula is known for the function $f(x)$, then the function may be keyed into program memory and numerically integrated by Simpson's rule. The user must specify the endpoints a and b of the interval over which integration is to be performed, and the number of subintervals n into which the interval (a, b) is to be divided. This n must be even; if it is not, Error will be displayed. The program will go on to compute $x_0 = a$, $x_j = x_0 + jh$, $j = 1, 2, ..., n-1$, and $x_n = b$ where

$$h = \frac{b - a}{n}$$

The integral $\int_a^b f(x)\, dx$ is approximated by equation (2) above, Simpson's rule.

17 program steps (or more) are available for user's function $f(x)$. Refer to the Instructions for keying in the function $f(x)$.

Remarks:

- Since there are actually 3 routines after LBL 1 for keying in the value of $f(x_j)$, one for $j = 0$, one for j odd, and one for j even, it is important that no other keys be pressed during the entry of the $f(x_j)$, lest the next $f(x_j)$ entered go into the wrong register.
- If n is not even erroneous results will occur.

01 *LBL1		50 ST+0			
02 CLRG	Input h	51 RCL2			
03 ST04		52 RCL1			
04 R/S		53 ST05			
05 ST09	– – – – – – – – –	54 –			
06 *LBL9	Input f_0	55 RCL3			
07 RCL3		56 ÷			
08 R/S	– – – – – – – – –	57 ST04			
09 ST01		58 0			
10 GSB6		59 ST09			
11 ENT↑		60 *LBL8	– – – – – – – – –		
12 +		61 GSB4			
13 ST+9	Input f_i for odd i	62 ST+0			
14 1		63 2			
15 ST+3		64 ST+9	$x \leftarrow x + h$		
16 RCL3		65 RCL3			
17 R/S	– – – – – – – – –	66 RCL9	$R_0 \leftarrow R_0 + 4\,f(x)$		
18 ST01		67 X=Y?	Exit for $R_9 = n$		
19 GSB6		68 GT05			
20 ST+9		69 GSB4			
21 1		70 GT08	– – – – – – – – –		
22 ST+3	Input f_i for even i	71 *LBL4			
23 GT09		72 RCL4			
24 *LBL2	– – – – – – – – –	73 ST+5			
25 3		74 RCL5	Subroutine		
26 RCL9		75 GSB0			
27 RCL1		76 GSB6			
28 –		77 RTN	– – – – – – – – –		
29 *LBL7	Calculate	78 *LBL5			
30 RCL4		79 3			
31 x	$\int f(x)\,dx$	80 RCL0	Exit		
32 X≷Y		81 GT07	– – – – – – – – –		
33 ÷	***	82 *LBL0			
34 RTN	– – – – – – – – –	83 RTN	For User's function		
35 *LBL6					
36 ENT↑					
37 +					
38 ST+0	Subroutine				
39 RTN	– – – – – – – – –				
40 *LBL3					
41 ST03					
42 R↓					
43 ST02					
44 R↓					
45 ST01	Input a, b, n				
46 GSB0	To calculate				
47 ST08					
48 RCL2	$\int_a^b f(x)\,dx$				
49 GSB0					

REGISTERS

0 Used	1 $f(x_i)$, a	2 b	3 n	4 h	5 x
6	7	8	9 Used	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program:			
	i for discrete case only: program step 1 to step 39.			
	ii for explicit functions only: program step 29 to step 83.			
2	For explicit functions, go to step 7, for discrete case, go to step 3.			
3	**Discrete Case:** input h	h	`GSB` `1`	h
4	Repeat this step for $j = 0, 1, ..., n$: Key in the function value at x_j	$f(x_j)$	`R/S`	j
5	Calculate the integral		`GSB` `2`	the integral
6	For a new case, go to step 3.			
7	**Explicit Function:** To key in your function f(x), first press, then switch to PRGM and key in f(x)		`GTO` `0`	
	Switch back to RUN*			
8	Input a, b, and n to calculate $\int_a^b f(x)dx$	a	`ENTER↑`	
		b	`ENTER↑`	
		n	`GSB` `3`	the integral
9	For a new set of a, b, and n, go to step 7.			
	*Note: Available program steps for f(x) are:			
	• 45 steps when only the EXPLICIT part is keyed in.			
	• 17 steps when both parts are keyed in.			

Example 1:

Given the values below for $f(x_j)$, $j = 0, 1, \ldots, 8$, calculate the approximations to the integral

$$\int_0^2 f(x)\, dx$$

by Simpson's formula.

The value for h is 0.25.

i	0	1	2	3	4	5	6	7	8
x_i	0	.25	.5	.75	1	1.25	1.5	1.75	2
$f(x_i)$	2	2.8	3.8	5.2	7	9.2	12.1	15.6	20

Keystrokes: **Outputs:**

(Key in the program from step 1 to step 39)

0.25 GSB ① ⟶   0.25

2 R/S 2.8 R/S 3.8 R/S 5.2
R/S 7 R/S 9.2 R/S 12.1
R/S 15.6 R/S 20 R/S ⟶   8.00

GSB ② ⟶   16.58   (the integral)

Example 2:

Find the value of

$$\int_0^{2\pi} \frac{dx}{1 - \cos x + 0.25}$$

for $n = 16$. Note that x is assumed to be in radians. For safety, if you work mostly in degrees, it is good programming practice to set the angular mode to radians at the beginning of the routine, then back to degrees at the end.

Keystrokes: **Outputs:**

(Key in the program from step 29 to step 83)

GTO ⓪ ,

Switch to PRGM

9 RAD f COS 1 x⪪y ⊟ .25
⊞ 9 ⅟x 9 DEG ,

Switch back to RUN

0 ENTER↑ 9 π 2 × 16
GSB ③ ⟶   8.36   (Answer)

IDEAL GAS EQUATION OF STATE

Many gases obey the ideal gas laws quite closely at reasonable temperatures and pressures. This program calculates any one of the four variables when data for the other three and the universal gas constant are entered. Likewise, the value of the universal gas constant can be determined by entering data for the four variables.

Equation:

$$PV = n\,RT$$

where: P is the absolute pressure

V is the volume

n is the number of moles present

R is the Universal Gas Constant

T is the absolute temperature

Table 1
Values of the Universal Gas Constant

Value of R	Units of R	Units of P	Units of V	Units of T
8.314	$N - m/g$ mole-$°K$	N/m^2	m^3/g mole	$°K$
83.14	$cm^3 - bar/g$ mole-$°K$	bar	cm^3/g mole	$°K$
82.05	$cm^3 - atm/g$ mole-$°K$	atm	cm^3/g mole	$°K$
0.08205	$ℓ - atm/g$ mole-$°K$	atm	$ℓ/g$ mole	$°K$
0.7302	atm-ft^3/lb mole-$°R$	atm	ft^3/lb mole	$°R$
10.73	psi-ft^3/lb mole-$°R$	psi	ft^3/lb mole	$°R$
1545	psf-ft^3/lb mole-$°R$	psf	ft^3/lb mole	$°R$

Remarks:

• At low temperatures or high pressures the ideal gas law does not represent the behavior of real gases.

• The value of R used must be compatible with the units of P, V, T.

• To ensure proper execution of the program initialize by pressing **GTO** [6] before entering data.

01 *LBL6			
02 0			
03 RTN			
04 *LBL1			
05 1			
06 GTO0			
07 *LBL2			
08 2			
09 GTO0			
10 *LBL3	Initialize and store data		
11 3			
12 GTO0			
13 *LBL4			
14 4			
15 GTO0			
16 *LBL5			
17 5			
18 *LBL0			
19 STO0			
20 R↓			
21 STOi			
22 X≠0?			
23 GTO6			
24 1			
25 STOi			
26 RCL1			
27 RCL2	PV		
28 ×			
29 RCL3			
30 RCL4	n RT		
31 ×			
32 RCL5			
33 ×			
34 GTOi			
35 *LBL1			
36 *LBL2	Calculate P or V		
37 X≷Y			
38 *LBL3			
39 *LBL4	Calculate n, R or T		
40 *LBL5			
41 ÷			
42 STOi	***		
43 RTN			

REGISTERS					
0 Indirect	1 P	2 V	3 n	4 R	5 T
6	7	8	9	.0	.1
.2	.3	.4	5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input four of the following:			
	absolute pressure	P	`GSB` `1`	0.00*
	volume	V	`GSB` `2`	0.00
	number of moles	n	`GSB` `3`	0.00
	universal gas constant	R	`GSB` `4`	0.00
	absolute temperature	T	`GSB` `5`	0.00
3	Calculate one of the following:**			
	absolute pressure	0.00	`GSB` `1`	P
	volume	0.00	`GSB` `2`	V
	number of moles	0.00	`GSB` `3`	n
	universal gas constant	0.00	`GSB` `4`	R
	absolute temperature	0.00	`GSB` `5`	T
4	For a new case, go to step 2 and change appropriate inputs.			
5	If program fails to execute properly press `GTO` `6` and start again.		`GTO` `6`	
*	Be sure that 0.00 is displayed after each data entry. If not press `GTO` `6` and reenter all data.			
**	Be sure 0.00 is displayed before `GSB` is executed to calculate unknown.			

Example 1:

0.63 moles of air are enclosed in 25000 cm³ of space at 1200°K. What is the pressure in bars? In atmospheres? Assume an ideal gas.

Keystrokes: **Outputs:**

25000 GSB ② 0.63 GSB ③
83.14 GSB ④ 1200 GSB ⑤
GSB ① ————————————————→   2.51   (bars)
82.05 GSB ④ GSB ① ————————→   2.48   (atm.)

Example 2:

What is the specific volume (ft^3/lb) of a gas at atmospheric pressure and a temperature of 513°R? The molecular weight is 29 lb/lb-mole.

Keystrokes: **Outputs:**

513 GSB ⑤ 29 g 1/x
GSB ③ 0.7302 GSB ④ 1
GSB ① GSB ② ————————————→   12.92   (ft^3/lb)

What is the density?

g 1/x FIX ③ ————————————————→   0.077   (lb/ft^3)

What is the density at 1.32 atmosphere and 555° R?

1.32 GSB ① 555 GSB ⑤ GSB ②
g 1/x ————————————————————→   0.094   (lb/ft^3)

RADIOACTIVE ISOTOPE DECAY

This program is designed to allow calculation of the decay in radioactivity of an isotope over a specified time interval. To use the program, select an isotope and key in its half-life. (Half-life data may be stored for up to 10 different isotopes in available storage registers.) Then key in two of the three variables:

A_0: Initial activity of the isotope.

t: Elapsed time.

A: Present activity.

The program then calculates the missing variable. Thus, for example, you are not restricted to finding the present activity, given time and initial activity; you may also solve for initial activity given time and present activity, or for time given initial activity and present activity.

The continuous memory feature of your calculator allows convenient storage and recall of the half-lifes of up to ten of the isotopes you most commonly use. Prior storage of the half-lifes eliminates having to enter them before each calculation and they are always available.

You may use any units for initial and present activity as long as they are consistent. The elapsed time *must* be input in the units: Days.Hours (DD.HH), where two full decimal places must be allotted to the hours. For instance an elapsed time of 5 days 18 hours would be keyed in and displayed as 5.18; a time of 1 day 6 hours as 1.06; and a time of 12 hours as 0.12.

Equations:

$$A = A_0 \left(\frac{1}{2} \right)^{t/\tau_{1/2}}$$

$$t = \frac{\tau_{1/2} \ln (A/A_0)}{\ln (1/2)}$$

where:

A_0 = initial radioactivity

A = present radioactivity

t = time elapsed, in hours

$\tau_{1/2}$ = half-life of radioisotope, in hours

Isotope	Half-Life in Hours $(\tau_{1/2})$
Cr^{51}	6672
Co^{57}	6480
Co^{60}	46460
I^{125}	1440
I^{131}	193.2
Cs^{137}	262980
H^3	107470
C^{14}	5.058×10^7
F^{18}	1.87
P^{32}	343.2
Se^{75}	2880
Sr^{85}	1536
In^{113}	1.73
Xe^{133}	126.5
Hg^{197}	65
Ra^{226}	1.3938×10^7

Remarks:

When recalling previously stored half-life data from the storage registers the program utilizes indirect addressing. Remember that the indirect addresses of storage registers .0 thru .5 are 10 thru 15 respectively.

If half-life of desired isotope has not previously been stored the user may key it in and store it in register 2, for use in the program.

Time is input and displayed in DD.HH format. To prevent "untidy" displays, such as 6.24 instead of 7.00 days, residual hours of 23.5 or greater are presented as 1.00 day.

The variable to be calculated is always input with a value of 0.00.

01 *LBL0	Recall selected isotope	
02 ST00	half-life	
03 RCLi		
04 ST02		
05 RTN		
06 *LBL1		
07 FIX2	Input data	
08 ST03		
09 R↓	Calculator determines which	
10 X=0?	variable is to be calculated	
11 ST04		
12 GSB9		
13 R↓		
14 ST05		
15 X=0?		
16 GTO8		
17 RCL4		
18 X=0?		
19 GTO7		
20 GTO6		
21 *LBL9		
22 ENT↑	Convert time from DD.HH	
23 INT	to hours	
24 2		
25 4		
26 x		
27 X⇄Y		
28 FRC		
29 EEX		
30 2		
31 x		
32 +		
33 ST04		
34 RTN		
35 *LBL8	Calculate A_0,	
36 RCL3	Initial activity	
37 RCL4		
38 RCL2		
39 ÷		
40 .		
41 5		
42 X⇄Y		
43 Y^x		
44 ÷		
45 ST05		
46 RTN		
47 *LBL7		
48 RCL3		
49 RCL5	Calculate t, time

50 ÷		
51 ST01		
52 LN		
53 RCL2		
54 x		
55 .		
56 5		
57 LN		
58 ÷		
59 2		
60 4		
61 ÷		
62 ENT↑	Convert from hours to	
63 INT	DD.HH	
64 X⇄Y		
65 FRC		
66 2		
67 4		
68 x		
69 2		
70 3		
71 .	If t ≥ 23.5 hours,	
72 5	round to nearest day	
73 X≤Y?		
74 GTO5		
75 R↓		
76 EEX		
77 2		
78 ÷		
79 +		
80 RTN		
81 *LBL6	Calculate A,	
82 .	Present activity	
83 5		
84 RCL4		
85 RCL2		
86 ÷		
87 Y^x		
88 RCL5		
89 x		
90 ST03		
91 RTN		
92 *LBL5		
93 R↓	Present t ≥ 23.5 hours	
94 R↓	as 1 day	
95 1		
96 +		
97 RTN		

REGISTERS

0 i	1 A/A_0	2 $\tau_{1/2}$	3 A	4 t	5 A_0
6 *	7	8	9	.0	.1
.2	3	4	5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*Registers 6 through .5 are available for isotope half life storage.

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
	To store half-lives of			
	commonly used isotopes.			
1'	Store half lifes of desired			
	isotopes in registers 6 through			
	9 and .0 through .5.*	$\tau_{1/2}$, hrs.	STO 6	
			:	
			STO 9	
			STO · 0	
			:	
			STO · 5	
	To calculate variables.			
2	Select desired isotope and			
	initialize by recalling its $\tau_{1/2}$ from			
	storage, using indirect address.	$\tau_{1/2}$ index	GSB 0	
2'	*or,* if isotope half life is not			
	stored, input $\tau_{1/2}$ manually.	$\tau_{1/2}$	STO 2	
3	Key in variables in this format:			
	• Activity at time zero	A_0	ENTER↑	
	• Elapsed time, in days hours			
	format	t, DD.HH	ENTER↑	
	• Present activity	A	GSB 1	unknown
	Important: Input zero for value			
	of unknown variable. Be sure			
	variables are entered in above			
	order.			
4	Other data may be recalled as			
	desired:			
	• Decay factor, A/A_0		RCL 1	
	• Half life, $\tau_{1/2}$		RCL 2	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
	• Present activity, A		RCL 3	
	• Elapsed time		RCL 4	
	• Initial activity		RCL 5	
	* Half lifes of up to 10 selected isotopes may be permanently stored in registers 6 through 9 and .0 through .5 having indirect addresses 6 through 15 respectively.			

Example:

An activity of 200 μCi is measured for a standard of Cr^{51} (with half-life 667.20 hours). What is the activity after a week?

Keystrokes: **Outputs:**

667.20 STO 2 $\longrightarrow$ 667.20   ($\tau_{1/2}$ for Cr^{51})

200 ENTER↑ $\longrightarrow$ 200.00   (A_o)

7 ENTER↑ $\longrightarrow$ 7.00   (t = 7 days)

0 GSB 1 $\longrightarrow$ 167.97   (A, μC_i)

(OR)

Calculate A_o given A = 167.97 (μC_i) and t = 7.00

0 ENTER↑ $\longrightarrow$ 0   Unknown

7 ENTER↑ $\longrightarrow$ 7.00   (t = 7 days)

167.97 GSB 1 $\longrightarrow$ 200.00   (A_o, μC_i)

ACID-BASE EQUILIBRIUM

This program calculates the hydrogen ion concentration, $[H_3O^+]$, and pH of a solution of a monoprotic weak acid if the ionization constant is known. Likewise, the program will calculate $[OH^-]$ concentration and pOH for solutions of weak bases given the ionization constant of the base. In addition, conversions from concentration to pH or pOH and vice versa and from pH to pOH, $[H_3O^+]$ to $[OH^-]$ etc. are included.

The following equation is used:

$$x^3 + K_a x^2 - (K_w + K_a C_a) x - K_w K_a = 0 \qquad (K_b \text{ For bases})$$

where:

$x = [H_3O^+]$ For acid, $[OH^-]$ For base

K_a = Ionization constant of acid $= \dfrac{[H_3O^+] \, [A^-]}{[HA]}$

K_b = Ionization constant of base $= \dfrac{[B^+] \, [OH^-]}{[BOH]}$

K_w = Ionization constant of water $= 10^{-14}$ @ 25° C

C_a or C_b = Concentration (moles/liter) of acid or base

The program uses Newton's method of approximating the solution of a polynomial where one evaluates f(x) successively with approximate values of X. First approximation of x is $x = (K_a C_a + K_w)^{1/2}$. Successive approximations are $x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$

The calculation is reiterated until x_{i+1} differs from x_i a small amount (1% or less).

References:

Butler, J.N., "Ionic Equilibrium, A Mathematical Approach", Addison-Wesley, 1964.

Dick, J.G., "Analytical Chemistry", McGraw-Hill, 1973.

This program is based upon a program submitted to the HP Users' Library by Alan J. Rubin.

01 *LBL1		50 X>Y?
02 STO0		51 GTO9
03 R↓		52 RCL2
04 STO1		53 RCL5
05 SCI2		54 -
06 RCL0		55 STO2
07 x		56 GTO0
08 EEX		57 *LBL9
09 CHS	Calculate and store con-	58 RCL2
10 1	stants	59 R/S
11 4		60 *LBL2
12 +		61 LOG
13 STO3		62 CHS
14 LSTX		63 FIX2
15 RCL1		64 RTN
16 x		65 *LBL3
17 STO4		66 CHS
18 RCL3		67 10ˣ
19 √x		68 SCI2
20 STO2		69 RTN
21 *LBL6		70 *LBL4
22 RCL1		71 1
23 +	Iteration sequence	72 4
24 RCL2		73 X≥Y
25 x		74 -
26 RCL3		75 FIX2
27 -		76 RTN
28 RCL2		77 *LBL5
29 x		78 EEX
30 RCL4		79 CHS
31 -		80 1
32 RCL2		81 4
33 3		82 X≥Y
34 x		83 ÷
35 RCL1		84 RTN
36 2		
37 x		
38 +		
39 RCL2		
40 x		
41 RCL3		
42 -		
43 ÷		
44 STO5		
45 ABS		
46 RCL2		
47 9		
48 9		
49 ÷	Test approximations	

Right column comments:
- Calculate next approximation
- Iterate
- *** Result [H₃O⁺] or [OH⁻]
- Convert Conc to pH or pOH or Kₐ, Kᵦ to pKₐ, pKᵦ ***
- Convert pH or pOH to Conc. or pKₐ, pKᵦ to Kₐ, Kᵦ ***
- Convert pH to pOH or vice versa ***
- Convert [H₃O⁺] → [OH⁻] or [OH⁻] → [H₃O⁺] ***

REGISTERS					
0 Cₐ or Cᵦ	1 K	2 X(est)	3 CK + Kw	4 K Kw	5 f(x)/f'(x)
6	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program:			
2	Enter ionization constant:			
	K_a or K_b	K	ENTER◆	K
	or pK_a or pK_b	pK	GSB 3	K
3	Input concentration (moles/liter)			
	of acid or base and calculate			
	conc. of $[H_3O^+]$ if acid (conc. of			
	$[OH^-]$ if base).	Conc.	GSB 1	$[H_3O^+]$ or $[OH^-]$
4	Convert concentration to pH or			
	pOH:			
	$[H_3O^+] \rightarrow$ pH	$[H_3O^+]$		
	or $[OH^-] \rightarrow$ pOH	or $[OH^-]$	R/S	pH or pOH
5	If desired the following con-			
	versions are available:			
	concentration of $[H_3O^+]$ or			
	$[OH^-]$ to pH or pOH	$[H_3O^+]$, $[OH^-]$	GSB 2	pH or pOH
	Ionization const. K_a or K_b to			
	pK_a or pK_b	K_a, K_b	GSB 2	pK_a, pK_b
	pH or pOH to concentration of			
	$[H_3O^+]$ or $[OH^-]$	pH, pOH	GSB 3	$[H_3O^+]$, $[OH^-]$
	pK to ionization constant	pK_a, pK_b	GSB 3	K_a, K_b
	pH to pOH or vice versa	pH, pOH	GSB 4	pOH, pH
	$[H_3O^+]$ to $[OH^-]$ or vice versa	$[H_3O^+]$, $[OH^-]$	GSB 5	$[OH^-]$, $[H_3O^+]$
6	If desired, error of calculation			
	may be reviewed.		RCL 5	f(x)/f'(x)

Example 1:

1.   Calculate the pH of a 1.0×10^{-4} molar solution of acetic acid if the ionization constant is 1.75×10^{-5}.

Keystrokes: **Outputs:**

1.75 EEX CHS 5 ENTER↑ EEX CHS

4 GSB ① ⎯⎯⎯⎯⎯⎯⎯⎯⎯→   3.41 −05   ($[H_3O^+]$)

R/S ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯→   4.47   (pH)

Example 2:

Calculate the pH of a sample of water containing 0.85 mg of ammonia as the only contaminant. K_b of ammonium hydroxide is 1.8×10^{-5} and the molecular wt. of ammonia is 17.

Keystrokes: **Outputs:**

1.8 EEX CHS ⑤ ENTER↑ 0.85

EEX CHS 3 ENTER↑ 17

÷ GSB ① ⎯⎯⎯⎯⎯⎯⎯⎯→   2.25 −05   ($[OH^-]$)

R/S ⎯⎯⎯⎯⎯⎯⎯⎯⎯→   4.65   (pOH)

GSB ④ ⎯⎯⎯⎯⎯⎯⎯⎯⎯→   9.35   (pH)

(Note: After entering ionization constant, calculate molar conc. of $NH_3 = 0.85 \times 10^{-3}/17 = 5 \times 10^{-5}M$).

Example 3:

Water in equilibrium with air contains carbon dioxide which forms a dilute solution of carbonic acid. If distilled water contains 1.35×10^{-5} moles/liter of carbon dioxide, what is the pH?

(The primary ionization constant of carbonic acid is 3.5×10^{-7}, the secondary ionization constant of 4.4×10^{-11} may be neglected).

Keystrokes: **Outputs:**

3.5 EEX CHS 7 ENTER↑ 1.35

EEX CHS 5 GSB ① ⎯⎯⎯⎯⎯→   2.03 −06   ($[H_3O^+]$)

R/S ⎯⎯⎯⎯⎯⎯⎯⎯⎯→   5.69   (pH)

(Examples 2 and 3 are taken from Kolthoff and Sandell, Textbook of Quantitative Inorganic Analysis, MacMillan, 1948).

CURVE FITTING

This program can be used to fit data to:

1. Straight lines (linear regression); $y = a + bx$.

2. Exponential curves; $y = ae^{bx}$ $(a > 0)$,

3. Logarithmic curves; $y = a + b \ln x$,

4. Power curves; $y = ax^b$ $(a > 0)$.

The regression coefficients a and b are found from solving the following equivalent of linear equations.

$$\begin{bmatrix} n & \Sigma X_i \\ \Sigma X_i & \Sigma X_i^2 \end{bmatrix} \begin{bmatrix} A \\ b \end{bmatrix} = \begin{bmatrix} \Sigma Y_i \\ \Sigma Y_i X_i \end{bmatrix}$$

While the relations of the variables are defined as the following:

Regression	A	X_I	Y_I	Code
Linear	a	x_i	y_i	5
Exponential	ln a	x_i	$\ln y_i$	6
Logarithmic	a	$\ln x_i$	y_i	7
Power	ln a	$\ln x_i$	$\ln y_i$	8

The coefficient of determination is:

$$r^2 = \frac{A\Sigma Y_i + b\Sigma X_i Y_i - \frac{1}{n}(\Sigma Y_i)^2}{\Sigma(Y_i^2) - \frac{1}{n}(\Sigma Y_i)^2}$$

The type of curve fit must be determined before data input begins, that is, by storing the code number into register 0.

Linear Regression
Code = 5

$y = a + bx$
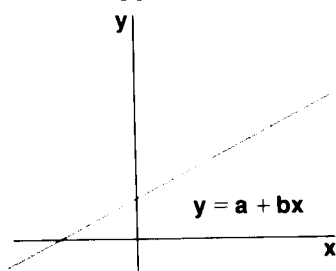
Exponential Curve Fit
Code = 6

$y = ae^{bx}$
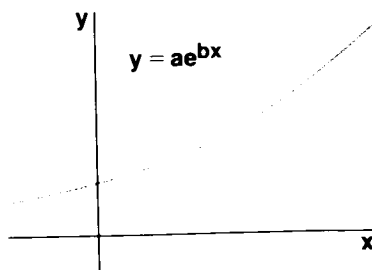
Logarithmic Curve Fit
Code = 7

$y = a + b \ln x$
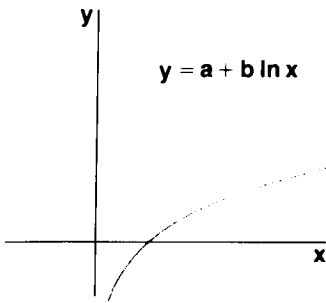
Power Curve Fit
Code = 8

$y = ax^b$
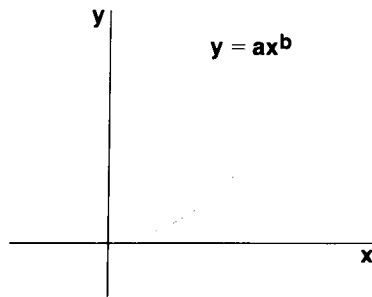
Remarks:

- Negative and zero values of x_i will cause a machine error for logarithmic curve fits. Negative and zero values of y_i will cause a machine error for exponential curve fits. For power curve fits both x_i and y_i must be positive, non-zero values.

- As the differences between x and/or y values become small, the accuracy of the regression coefficients will decrease.

01 *LBL1			50 x	Determinate	
02 X≠Y			51 STO7		
03 GSB1	Input data		52 R↓		
04 Σ+			53 x		
05 RTN	– – – – – – –		54 RCL7		
06 *LBL7			55 –		
07 LN	Log		56 RTN		
08 RTN			57 *LBL3		
09 *LBL8	– – – – – – –		58 RCL4		
10 LN			59 RC.3		
11 *LBL6	Power and exp.		60 x		
12 X≠Y			61 RCL5		
13 LN			62 RC.5		
14 X≠Y			63 x		
15 RTN	– – – – – – –		64 ÷		
16 *LBL2			65 RC.3		
17 RC.0			66 X²		
18 RC.2			67 RC.0	Calculate r^2	
19 RC.1			68 ÷		
20 RC.1			69 STO9		
21 GSB9			70 –		
22 STO3			71 RC.4		
23 RC.3			72 RCL9		
24 RC.2			73 –		
25 RC.1			74 ÷		
26 RC.5			75 R/S		
27 GSB9			76 *LBL4		
28 RCL3	Calculate A, b, and a, b		77 GTO1		
29 ÷			78 *LBL8	– – – – – – –	
30 STO4			79 RCL5		
31 GSB1			80 Yˣ		
32 STO6			81 GTO5		
33 R/S			82 *LBL6		
34 RC.0			83 RCL5		
35 RC.5			84 x		
36 RC.1			85 eˣ		
37 RC.3			86 *LBL9		
38 GSB9			87 RCL6		
39 RCL3			88 x	Input x to calculate y	
40 ÷			89 RTN		
41 STO5			90 *LBL7		
42 RTN			91 LN		
43 *LBL6	Inverse transform		92 *LBL5		
44 *LBL8			93 RCL5		
45 eˣ			94 x		
46 *LBL5			95 RCL6		
47 *LBL7			96 +		
48 RTN	– – – – – – –		97 RTN		
49 *LBL9					

REGISTERS					
0 Index	1 x	2 y	3 det	4 A	5 b
6 a	7 Used	8	9 $1/n\,(\Sigma Y)^2$	.0 n	.1 ΣX
.2 ΣX^2	.3 ΣY	.4 ΣY^2	.5 ΣXY	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Initialize.		**f** [REG]	
3	Store curve fit code (5 or 6 or 7 or 8) in register 0	code	[STO] [0]	code
4	(Repeat for i = 1, 2, ..., n.) Input x_i value and y_i value.	x_i	[ENTER♦]	
		y_i	[GSB] [1]	i
5	Calculate regression coefficients		[GSB] [2]	a
			[R/S]	b
6	Calculate r^2.		[GSB] [3]	r^2
7	(Repeat if necessary.) Input x to calculate $\hat{y}$.	x	[GSB] [4]	$\hat{y}$
8	For a new case, go to step 2.			

Example 1:

(Linear, code = 5):

x_i	40.5	38.6	37.9	36.2	35.1	34.6
y_i	104.5	102	100	97.5	95.5	94

Solution:

$a = 33.53$, $b = 1.76$

$r^2 = 0.99$

i.e., $y = 33.53 + 1.76\ x$

For $x = 37$, $\hat{y} = 98.65$

Keystrokes: Outputs:

f [REG] 5 [STO] [0] ———————➤   5.00

40.5[ENTER♦] 104.5 [GSB] [1]

38.6[ENTER♦] 102 [GSB] [1]

37.9[ENTER♦] 100 [GSB] [1]

36.2[ENTER♦] 97.5 [GSB] [1]

35.1 ENTER♦ 95.5 GSB ①
34.6 ENTER♦ 94 GSB ① ⟶   6.00
GSB ② ⟶   33.53   (a)
R/S ⟶   1.76   (b)
GSB ③ ⟶   0.99 (r^2)
37 GSB ④ ⟶   98.65 $(\hat{y})$

Example 2:
(Exponential, Code = 6):

x_i	.72	1.31	1.95	2.58	3.14
y_i	2.16	1.61	1.16	.85	0.5

Solution:

$a = 3.45$, $b = -0.58$

$y = 3.45\ e^{-0.58x}$

$r^2 = 0.98$

For $x = 1.5$, $\hat{y} = 1.44$

Example 3:
(Logarithmic, Code = 7):

x_i	3	4	6	10	12
y_i	1.5	9.3	23.4	45.8	60.1

Solution:

$a = -47.02$, $b = 41.39$

$y = -47.02 + 41.39 \ln x$

$r^2 = 0.98$

For $x = 8$, $\hat{y} = 39.06$

For $x = 14.5$, $\hat{y} = 63.67$

Example 4:
(Power, Code = 8):

x_i	10	12	15	17	20	22	25	27	30	32	35
y_i	0.95	1.05	1.25	1.41	1.73	2.00	2.53	2.98	3.85	4.59	6.02

Solution:

$a = .03$, $b = 1.46$

$y = .03x^{1.46}$

$r^2 = 0.94$

For $x = 18$, $\hat{y} = 1.76$

For $x = 23$, $\hat{y} = 2.52$

NORMAL AND INVERSE NORMAL DISTRIBUTION

This program evaluates the standard normal density function f(x) and the normal integral Q(x) for given x. If Q is given, x can also be found.

The standard normal distribution has mean 0 and standard deviation 1.

Equations:

1. Standard normal density

$$f(x) = \frac{1}{\sqrt{2\pi}} \; e^{-\frac{x^2}{2}}$$
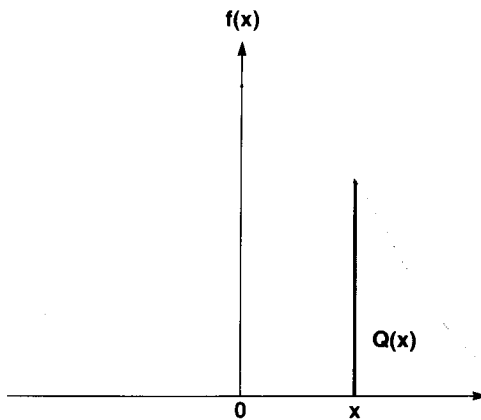
2. Normal integral

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$$

Polynomial approximation is used to compute Q(x) for given x.

Define $R = f(x) (b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5) + \epsilon(x)$

where:

$$|\epsilon(x)| < 7.5 \times 10^{-8}$$

$$t = \frac{1}{1 + r \, |x|}, \qquad r = 0.2316419$$

$$b_1 = .319381530$$
$$b_2 = -.356563782$$
$$b_3 = 1.781477937$$
$$b_4 = -1.821255978$$
$$b_5 = 1.330274429$$

Then $Q(x) = \begin{cases} R & \text{if } x \geq 0 \\ 1 - R & \text{if } x < 0 \end{cases}$

3. Inverse normal

For a given $Q > 0$, x can be found such that

$$Q = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$$

The following rational approximation is used:

Define $y = t - \dfrac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3} + \epsilon(Q)$

where:

$$\left| \epsilon(Q) \right| < 4.5 \times 10^{-4}$$

$$t = \begin{cases} \sqrt{\ln \dfrac{1}{Q^2}} & \text{if } 0 < Q \leq 0.5 \\[3mm] \sqrt{\ln \dfrac{1}{(1-Q)^2}} & \text{if } 0.5 < Q < 1 \end{cases}$$

$c_0 = 2.515517$	$d_1 = 1.432788$
$c_1 = 0.802853$	$d_2 = 0.189269$
$c_2 = 0.010328$	$d_3 = 0.001308$

Then $x = \begin{cases} y & \text{if } 0 < Q \leq 0.5 \\ -y & \text{if } 0.5 < Q < 1 \end{cases}$

Reference:

Abramowitz and Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, 1970.

01 *LBL1		50 ENT↑	
02 STO7		51 RCL5	
03 X²		52 x	
04 2		53 RCL4	
05 ÷		54 GSB7	
06 CHS		55 RCL3	
07 eˣ		56 GSB7	
08 Pi		57 RTN	
09 2	Calculate f(x)	58 *LBL3	
10 x		59 STO7	
11 √x̄		60 .	
12 ÷		61 5	
13 STO9		62 X≷Y	
14 RTN		63 X>Y?	
15 *LBL2		64 GSB8	
16 GSB1		65 X²	
17 1		66 1/X	
18 RCL8		67 LN	
19 RCL7		68 √x̄	
20 ABS		69 STO8	
21 x		70 GSB6	
22 +		71 1	
23 1/X	Calculate Q (x)	72 +	
24 GSB6		73 STO9	
25 RCL2		74 CLX	
26 GSB7		75 RCL2	Calculate X
27 RCL1		76 x	
28 GSB7		77 RCL1	
29 RCL9		78 GSB7	
30 x		79 RCL8	
31 RCL7		80 +	
32 X≤0?		81 RCL9	
33 GTO9		82 ÷	
34 X≷Y		83 −	
35 RTN		84 STO6	
36 *LBL9		85 RCL7	
37 X≷Y		86 .	
38 *LBL8		87 5	
39 1		88 X≷Y	
40 −		89 X>Y?	
41 CHS		90 GTO5	
42 RTN		91 RCL6	
43 *LBL7	Subroutines	92 RTN	
44 +		93 *LBL5	
45 x		94 RCL6	
46 RTN		95 CHS	
47 *LBL6		96 RTN	
48 ENT↑			
49 ENT↑			

REGISTERS

0 r, C_0	1 b_1, C_1	2 b_2, C_2	3 b_3, d_1	4 b_4, d_2	5 b_5, d_3
6 y	7 x, Q	8 t	9 f(x), deno.	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
	I. Normal distribution: from			
	program step 1 to step 57.			
	ii. Inverse normal distribution:			
	from program step 38 to step 96.			
	iii. Both: the entire program.			
2	For normal distribution, go to			
	step 3, for inverse, go to step 7.			
3	Store constants for normal			
	distribution.	r	STO 0	r
		b_1	STO 1	b_1
		b_2	STO 2	b_2
		b_3	STO 3	b_3
		b_4	STO 4	b_4
		b_5	STO 5	b_5
4	Optional: Input x to calculate f(x)	x	GSB 1	f(x)
5	Input x to calculate Q(x)	x	GSB 2	Q(x)
6	For a new x, go to step 4 or			
	step 5.			
7	Store constants for inverse	c_0	STO 0	c_0
		c_1	STO 1	c_1
		c_2	STO 2	c_2
		d_1	STO 3	d_1
		d_2	STO 4	d_2
		d_3	STO 5	d_3
8	Input Q(x) to calculate x.	Q(x)	GSB 3	x
9	For a new Q(x), go to step 8.			

Example 1:

(Normal distribution):

Find f(x) and Q(x) for x = 1.18 and x = –2.28

Keystrokes: **Outputs:**

(Key in the program as shown in the Instructions)

0.2316419 **STO** ⓪ 0.31938153
STO ① 0.356563782 **CHS** **STO**
② 1.781477937 **STO** ③
1.821255978 **CHS** **STO** ④
1.330274429 **STO** ⑤

1.18 **GSB** ① ⟶	0.20	(f(1.18))
1.18 **GSB** ② ⟶	0.12	(Q(1.18))
2.28 **CHS** **GSB** ② ⟶	0.99	(Q(−2.28))
2.28 **CHS** **GSB** ① ⟶	0.03	(f(−2.28))

Example 2:

(Inverse):

Given Q = 0.12 and Q = 0.95, find x's

Keystrokes: **Outputs:**

(Key in program as shown in the Instructions)

2.515517 **STO** ⓪ 0.802853
STO ① 0.010328 **STO** ②
1.432788 **STO** ③ 0.189269
STO ④ 0.001308 **STO** ⑤

0.12 **GSB** ③ ⟶	1.18	(x)
0.95 **GSB** ③ ⟶	−1.65	(x)

FACTORIAL, PERMUTATION AND COMBINATION

$$\text{Factorial} \quad n! = n\,(n-1)\,(n-2)\,\cdots\,2\cdot 1$$

$$\text{Permutation} \quad {}_mP_n = \frac{m!}{(m-n)!} = m(m-1)\ldots(m-n+1)$$

$$\text{Combination} \quad {}_mC_n = \frac{m!}{(m-n)!\ n!} = \frac{m(m-1)\ldots(m-n+1)}{1\cdot 2\cdot \ldots \cdot n}$$

where m, n are integers and $0 \leqslant n \leqslant m$.

Remarks:

This program will compute factorials for positive integers between 2 and 69.

$$n! = n\,(n-1)\,(n-2)\,\ldots\,(2)\,(1)$$

For large values of n, the program will take some time to arrive at a result, up to a maximum of about 20 seconds for $n = 69$.

The program does not check input values and will return incorrect answers for values of $n < 2$ or $n > 69$ or non-integer n.

${}_mP_0 = 1$, ${}_mP_1 = m$, ${}_mP_m = m!$ Therefore n! should be used for large m.

${}_mC_0 = {}_mC_m = 1$

${}_mC_1 = {}_mC_{m-1} = m$

${}_mC_n = {}_mC_{m-n}$

		Calculate n!			Calculate $_mC_n$
01 *LBL1			50 GT08		
02 1			51 -		
03 ST.0			52 LSTX		
04 X≷Y			53 X≤Y?		
05 *LBL4			54 GSB9		
06 Sx.0			55 ST01		
07 :			56 1		
08 -			57 ST00		
09 X≠Y?			58 +		
10 GT04			59 ST02		
11 RC.0			60 CLX		
12 RTN			61 X=Y?		
13 *LBL2			62 GT07		
14 X>Y?			63 *LBL0		
15 GT08			64 R↓		
16 ENT↑			65 ISZ		
17 0			66 RCL0		
18 X=Y?			67 X>Y?		
19 GT07			68 GT08		
20 CLX			69 RCL1		
21 :			70 X≷Y		
22 X=Y?			71 +		
23 ST06			72 LSTX		
24 -			73 ÷		
25 ST08			74 RCL2		
26 R↓			75 x		
27 ST01			76 ST02		
28 *LBL5			77 GT08		
29 RCL1			78 *LBL9		
30 :			79 ST02		
31 -			80 X≷Y		
32 ST01			81 RTN		
33 x			82 *LBL8		
34 DSZ			83 RCL2		
35 GT05			84 RTN		
36 RTN					
37 *LBL6					
38 R↓					
39 R↓					
40 RTN					
41 *LBL7					
42 ENT↑					
43 :					
44 RTN					
45 *LBL8					
46 0					
47 ÷					
48 *LBL3					
49 X>Y?					

Calculate n! ***

Calculate $_mP_n$ ***

$_mP_1 = m$

*** 1

Error

$_mC_n = {_mC_{m-n}}$ ***

REGISTERS

0 n!	1 m,	2 Used	3	4	5
6	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
	i. Factorial: from program step 1 to step 12.			
	ii. Permutation: from program step 12 to step 47.			
	iii. Combination: from program step 41 to step 84.			
2	For factorial, go to step 3, for permutation go to step 5, for combination go to step 7.			
3	Input n to calculate n!	n	GSB 1	n!
4	Repeat step 3 for another n.			
5	Input m and n to calculate $_mP_n$	m	ENTER↑	
		n	GSB 2	$_mP_n$
6	Repeat step 5 for a different set of m and n.			
7	Input m and n to calculate $_mC_n$	m	ENTER↑	
		n	GSB 3	$_mC_n$
8	Repeat step 7 for a different set of m and n.			

Example 1:

(Factorial):

Find n! for n = 5 and n = 10

Keystrokes: Outputs:

(Key in the program as shown in the Instructions)

5 GSB 1 ————————————→   120.00   (5!)
10 GSB 1 ————————————→   3628800.00   (10!)

Example 2:

(Permutation):

Find $_{43}P_3$ and $_{73}P_4$.

Keystrokes: Outputs:

(Key in the program as shown in the Instructions)

43 ENTER↑ 3 GSB 2 ───────⟶   74046.00 $(_{43}P_3)$

73 ENTER↑ 4 GSB 2 ───────⟶   26122320.00 $(_{73}P_4)$

### Example 3:

(Combination):

Find $_{73}C_4$ and $_{43}C_3$.

Keystrokes: Outputs:

(Key in the program as shown in the Instructions)

73 ENTER↑ 4 GSB 3 ───────⟶   1088430.00 $(_{73}C_4)$

43 ENTER↑ 3 GSB 3 ───────⟶   12341.00 $(_{43}C_3)$

STATIC EQUILIBRIUM AT A POINT

This program calculates the two reaction forces necessary to balance a given two-dimensional force vector. The direction of the reaction forces may be specified as a vector of arbitrary length or by Cartesian coordinates using the point of force application as the origin.
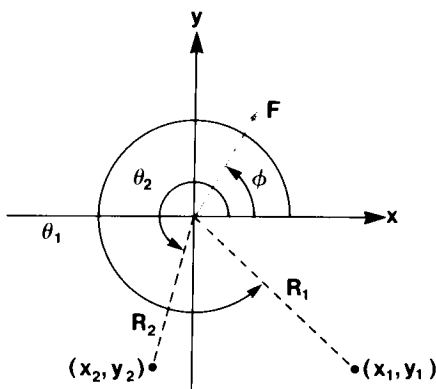
Equations:

$$R_1 \cos \theta_1 + R_2 \cos \theta_2 = F \cos \phi$$
$$R_1 \sin \theta_1 + R_2 \sin \theta_2 = F \sin \phi$$

where:

F is the known force;

ϕ is the direction of the known force;

R_1 is one reaction force;

θ_1 is the direction of R_1;

R_2 is the second reaction force;

θ_2 is the direction of R_2;

The coordinates x_1 and y_1 are referenced from the point where F is applied to the end of the member along which R_1 acts; x_2 and y_2 are the coordinates referenced from the point where F is applied to the end of the member along which R_2 acts.

Remarks:

This program assumes the calculator is set in DEG mode.

01 *LBL1	Input y_1, x_1	50 -	
02 →P	_ _ _ _ _ _ _ _ _ _ _ _	51 RCL6	
03 X≠Y		52 ÷	•••
04 *LBL2		53 RTN	
05 1	Input θ_1, and store sin θ_1,		
06 →R	cos θ_1		
07 STO0			
08 X≠Y			
09 STO1			
10 RTN	_ _ _ _ _ _ _ _ _ _ _ _		
11 *LBL3	Input y_2, x_2		
12 →F	_ _ _ _ _ _ _ _ _ _ _ _		
13 X≠Y			
14 *LBL4			
15 1			
16 →R	Input θ_2 and store sin θ_2,		
17 STO2	cos θ_2		
18 X≠Y			
19 STO3	_ _ _ _ _ _ _ _ _ _ _ _		
20 RTN			
21 *LBL5			
22 →R			
23 STO4			
24 X≠Y			
25 STO5			
26 RCL4			
27 RCL3			
28 ×			
29 RCL5			
30 RCL2			
31 ×			
32 -			
33 RCL1	Input ϕ and F and calculate		
34 RCL2	reaction forces		
35 ×			
36 RCL0			
37 RCL3			
38 ×			
39 -			
40 ÷			
41 R/S	•••		
42 LSTX			
43 STO6			
44 RCL5			
45 RCL0			
46 ×			
47 RCL4			
48 RCL1			
49 ×			

REGISTERS

0 cos θ_1	1 sin θ_1	2 cos θ_2	3 sin θ_2	4 F cos ϕ	5 F sin ϕ
6 Used	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Define reaction directions as			
	Cartesian coordinates or as			
	vectors of arbitrary magnitude.			
	(Use the point of force appli-			
	cation as the origin):			
	Define direction one in rec-			
	tangular form	y_1	ENTER♦	y_1
		x_1	GSB 1	$\sin \theta_1$
	or in polar form	θ_1	GSB 2	$\sin \theta_1$
	and			
	Define direction two in			
	rectangular form	y_2	ENTER♦	
		x_2	GSB 3	$\sin \theta_2$
	or in polar form	θ_2	GSB 4	$\sin \theta_2$
3	Key in known force: direction			
	then magnitude and compute			
	reactions.	ϕ	ENTER♦	
		F	GSB 5	R_1
			R/S	R_2
4	To change force, go to step 3.			
	To change either or both reac-			
	tion directions, go to step 2.			

Example 1:

Find the reaction forces in the pin-jointed structure shown below.
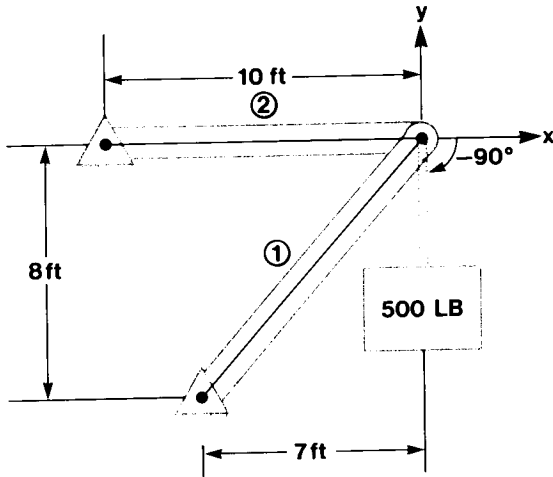
Keystrokes: **Outputs:**

8 CHS ENTER↑ 7 CHS

GSB 1 ─────────────────────────▶   −0.75

0 ENTER↑ 10 CHS GSB 3 ─────▶   0.00

90 CHS ENTER↑ 500 GSB 5 ─────▶   −664.38   (R₁)

R/S ───────────────────────────▶   437.50   (R₂)

Example 2:

Find the reaction forces for the diagram below:

Keystrokes: **Outputs:**

30 GSB 2 ——————————→   0.50

125 GSB 4 —————————→   0.82
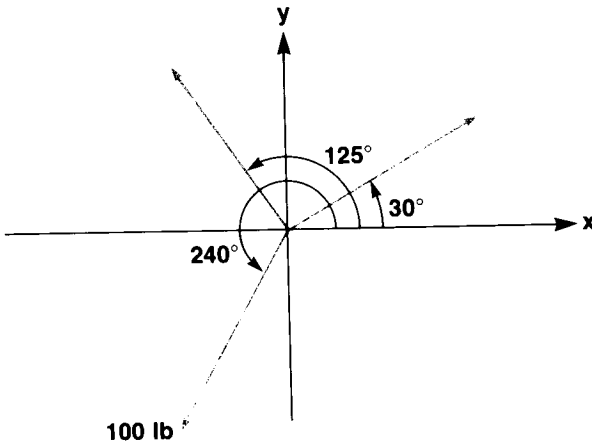
240 ENTER↑ 100 GSB 5 ———→   90.98   (R_1)

R/S ———————————————→   50.19   (R_2)

SECTION PROPERTIES

The properties of arbitrarily shaped sections which are composed of rectangles can be evaluated using this program.

The program calculates the area of the section, the centroid of the area, the moments of inertia about any specified set of axes, the polar moment of inertia about the specified axis, the moments of inertia about an axis translated to the centroid, the moments of inertia of the principal axis, and the rotation angle between the translated axis and the principal axis.

Equations:

$$A_{si} = \Delta x_i \, \Delta y_i$$

$$A = S_{s1} + A_{s2} + A_{s3} + \ldots + A_{sn}$$

$$\bar{x} = \frac{\sum\limits_{i=1}^{n} x_{oi} \, A_{si}}{A}$$

$$\bar{y} = \frac{\sum\limits_{i=1}^{n} y_{oi} \, A_{si}}{A}$$

$$I_x = \sum_{i=1}^{n} \left(y_{oi}^2 + \frac{\Delta y_i^2}{12} \right) A_{si}$$

$$I_y = \sum_{i=1}^{n} \left(x_{oi}^2 + \frac{\Delta x_i^2}{12} \right) A_{si}$$

$$J = I_x + I_y$$

$$I_{xy} = \sum_{i=1}^{n} x_{oi} \, y_{oi} \, A_{si}$$

$$I_{\bar{x}} = I_x - A\bar{y}^2$$

$$I_{\bar{x}\bar{y}} = I_{xy} - A\bar{x}\bar{y}$$

$$I_{\bar{y}} = I_y - A\bar{x}^2$$

$$\phi = \frac{1}{2} \tan^{-1} \frac{-2 \, I_{\bar{x}\bar{y}}}{I_{\bar{x}} - I_{\bar{y}}}$$

where:

Δx_i is the width of a rectangular element;

Δy_i is the height of a rectangular element;

A_{si} is the area of an element;

A is the total area of the section;

$\bar{x}$ is the x coordinate of the centroid;

$\bar{y}$ is the y coordinate of the centroid;

x_{oi} is the x coordinate of the centroid of an element;

y_{oi} is the y coordinate of the centroid of an element;

I_x is the moment of inertia about the x-axis;

I_y is the moment of inertia about the y-axis;

J is the moment of inertia about the origin;

I_{xy} is the product of inertia;

$I_{\bar{x}}$ is the moment of inertia about the x-axis translated to the centroid;

$I_{\bar{y}}$ is the moment of inertia about the y-axis translated to the centroid;

$I_{\bar{x}\bar{y}}$ is the product of inertia about the translated axis;

ϕ is the angle between the translated axis and the principal axis;

Reference:

Wojciechowski, Felix; ''Properties of Plane Cross Sections''; *Machine Design;* P. 105, Jan 22, 1976.

01 *LBL8		
02 STO4		
03 R↓		
04 STO3		
05 R↓		
06 STO2		
07 X≷Y	Input Δx, Δy, x_{oi}, and y_{oi}	
08 STO1	and calculate	
09 x		
10 ST.5		
11 ENT↑		
12 ST+0		
13 RCL3		
14 x		
15 ST+5		
16 R↓		
17 RCL4		
18 x		
19 ST+6		
20 RCL2		
21 X²		
22 ÷		
23 2		
24 ÷		
25 RCL4		
26 X²		
27 +		
28 RC.5		
29 x		
30 ST+7		
31 RCL1		
32 X²		
33 ÷		
34 2		
35 ÷		
36 RCL3		
37 X²		
38 +		
39 RC.5		
40 x		
41 ST+8		
42 RCL3		
43 RCL4		
44 x		
45 RC.5		
46 x		
47 ST+9		
48 RTN		
49 *LBL1		

50 RCL5		
51 RCL0		
52 ÷	Calculate $\bar{x}$, $\bar{y}$, and A	
53 RTN		
54 *LBL9		
55 RCL6		
56 RCL0		
57 ÷		
58 RTN		
59 RCL8		
60 RTN		
61 *LBL2		
62 RCL7		
63 GSB9		
64 X²		
65 RCL0		
66 x		
67 -		
68 ST.2	Calculate $I_{\bar{x}}$, $I_{\bar{y}}$, and $I_{\bar{x}\bar{y}}$	
69 R/S		
70 RCL8		
71 GSB1		
72 X²		
73 RCL0		
74 x		
75 -		
76 ST.3		
77 R/S		
78 RCL9		
79 GSB1		
80 GSB9		
81 x		
82 RCL0		
83 x		
84 -		
85 ST.4		
86 RTN		
87 *LBL3		
88 RC.4		
89 2		
90 x		
91 RC.3	Calculate ϕ	
92 RC.2		
93 -		
94 ÷		
95 TAN⁻¹		
96 2		
97 ÷		
98 RTN		

REGISTERS

0 ΣA	1 Δx_i	2 Δy_i	3 x_{oi}	4 y_{oi}	5 $\Sigma x_{oi}\, A_{si}$
6 $\Sigma y_{oi}\, A_{si}$	7 ΣI_x	8 ΣI_y	9 ΣI_{xy}	.0	.1
.2 $I_{\bar{x}}$	.3 $I_{\bar{y}}$	.4 $I_{\bar{x}\bar{y}}$	.5 A_{si}	16	17
18	19	20	21	22	23
24	25	26	27	28	29

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Initialize		▨ REG	
3	Input Δx, Δy, x_{oi}, y_{oi}	Δx	ENTER ↓	
		Δy	ENTER ↓	
		x_{oi}	ENTER ↓	
		y_{oi}	GSB 0	
4	Repeat step 3 for more sections.			
5	To calculate $\bar{x}$, $\bar{y}$, A		GSB 1	$\bar{x}$
			R/S	$\bar{y}$
			R/S	A
6	Optional: To recall I_x		RCL 7	I_x
	I_y		RCL 8	I_y
	I_{xy}		RCL 9	I_{xy}
7	To calculate $I_{\bar{x}}$		GSB 2	$I_{\bar{x}}$
	$I_{\bar{y}}$		R/S	$I_{\bar{y}}$
	$I_{\bar{x}\bar{y}}$		R/S	$I_{\bar{x}\bar{y}}$
8	To calculate ϕ		GSB 3	ϕ
9	For a new case, go to step 2.			

Example 1:

Given the rectangle below, find $\bar{x}$, $\bar{y}$, A, I_x, I_y, I_{xy}, $I_{\bar{x}}$, $I_{\bar{y}}$, $I_{\bar{x}\bar{y}}$ and ϕ.
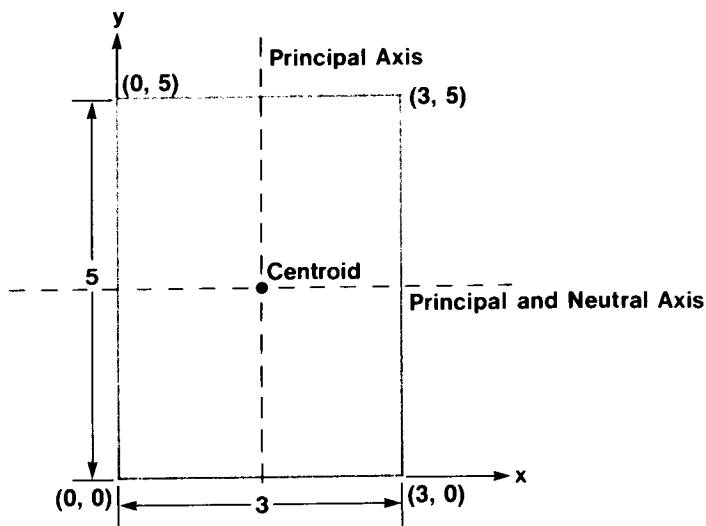
TABLE OF INPUTS

Section	Δx	Δy	x_o	y_o
1	3	5	1.5	2.5

Keystrokes:　　　　　　　　　　　**Outputs:**

Keystrokes	Output	
f **REG** 3 **ENTER↑** 5 **ENTER↑** 1.5		
ENTER↑ 2.5 **GSB** **0** ⟶	56.25	
GSB **1** ⟶	1.50	($\bar{x}$)
R/S ⟶	2.50	($\bar{y}$)
R/S ⟶	15.00	(A)
RCL **7** ⟶	125.00	(I_x)
RCL **8** ⟶	45.00	(I_y)
RCL **9** ⟶	56.25	(I_{xy})
GSB **2** ⟶	31.25	($I_{\bar{x}}$)
R/S ⟶	11.25	($I_{\bar{y}}$)
R/S ⟶	0.00	($I_{\bar{x}\bar{y}}$)
GSB **3** ⟶	0.00	(ϕ)

Example 2:

Calculate the section properties for the beam shown below.
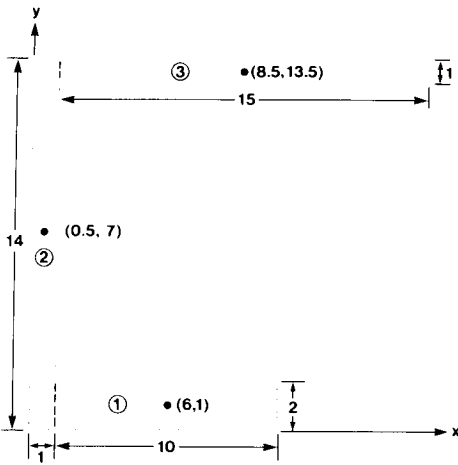
TABLE OF INPUTS

Section	Δx	Δy	x_{ol}	y_{ol}
1	10	2	6	1
2	1	14	0.5	7
3	15	1	8.5	13.5

Keystrokes:	Outputs:	
f REG 10 ENTER↑ 2 ENTER↑ 6 ENTER↑ 1 GSB 0 ——————→	120.00	
1 ENTER↑ 14 ENTER↑ 0.5 ENTER↑ 7 GSB 0 ——————→	49.00	
15 ENTER↑ 1 ENTER↑ 8.5 ENTER↑ 13.5 GSB 0 ——————→	1721.25	
GSB 1 ——————→	5.19	$(\bar{x})$
R/S ——————→	6.54	$(\bar{y})$
R/S ——————→	49.00	(A)
RCL 7 ——————→	3676.33	(I_x)
RCL 8 ——————→	2256.33	(I_y)
RCL 9 ——————→	1890.25	(I_{xy})
GSB 2 ——————→	1580.00	$(I_{\bar{x}})$
R/S ——————→	934.49	$(I_{\bar{y}})$
R/S ——————→	225.61	$(I_{\bar{x}\bar{y}})$
GSB 3 ——————→	−17.48	(ϕ)

FIELD ANGLE OR BEARING TRAVERSE

This program uses angles and/or deflections turned from a reference azimuth and horizontal distances, or quadrant bearings and horizontal distances, to compute the coordinates of successive points in a traverse. For a closed traverse, the area enclosed and closure distance and azimuth are computed.

(Note: Angles left and deflections left must be entered as negative numbers.)

Equations:

$$N_{i+1} = N_1 + HD \cos Az$$

$$E_{i+1} = E_1 + HD \sin Az$$

$$Area = \sum_{k=1}^{n} LAT_k \left(\tfrac{1}{2} DEP_k + \sum_{j=1}^{k-1} DEP_j \right)$$

where:

$$DEP_k = E_{k+1} - E_k \text{ and } LAT_k = N_{k+1} - N_k$$

01 *LBL1	Store starting point	50 2		
02 FIX4	coordinates and 180°	51 ÷		
03 CLRG		52 RCL7		
04 STO1		53 -		
05 X≷Y		54 x		
06 STO2		55 ST+8		
07 1		56 RCL6		
08 8		57 RCL2		
09 0		58 +		
10 STO3		59 R/S	***	
11 R/S		60 RCL7		
12 →H		61 RCL1		
13 RCL3	Reference azimuth	62 +		
14 →H		63 R/S	***	
15 -		64 *LBL4		
16 GTO8		65 X≷Y		
17 *LBL2		66 STO9		
18 →H	Angle input	67 X≷Y	Convert bearing and	
19 RCL3		68 ENT↑	quadrant code to azimuth.	
20 →H		69 ENT↑		
21 +		70 2		
22 →HMS		71 ÷		
23 *LBL3	Deflection angle input	72 INT		
24 →H		73 RCL3		
25 RCL4		74 x		
26 +		75 X≷Y		
27 *LBL0		76 RCL3		
28 :		77 x		
29 →R		78 COS		
30 →P	Compute azimuth	79 RCL9		
31 *LBL9		80 →H		
32 X≷Y		81 x		
33 X>0?		82 -		
34 GTO8		83 GTO8		
35 3		84 *LBL5	Area	
36 6		85 RCL8		
37 0		86 ABS		
38 +		87 R/S	***	
39 *LBL6		88 RCL7		
40 STO4		89 RCL6		
41 →HMS		90 →P		
42 R/S	***	91 R/S	Setup for closure	
43 ST+5	Input horizontal distance	92 GTO9	***	
44 RCL4				
45 X≷Y				
46 →R				
47 ST+6				
48 X≷Y				
49 ST+7	Compute next coord. and			
	accumulate area.			

REGISTERS

0	1 Beg. E	2 Beg. N	3 180	4 Az	5 Σ HD
6 Lat.	7 Dep.	8 Area	9 Bearing	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

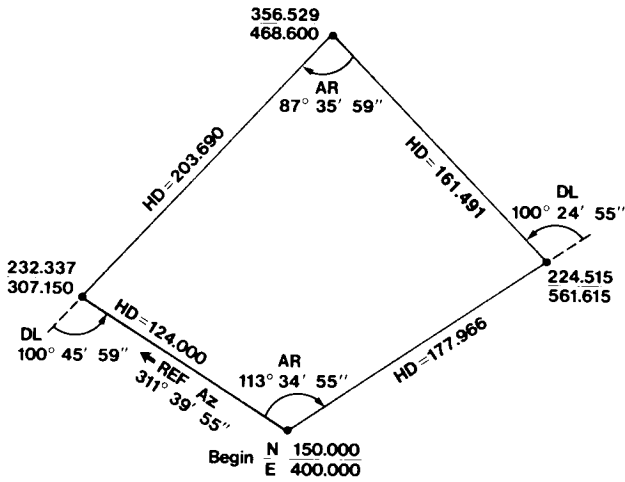*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Key in beginning coordinates	BEG N	ENTER♦	
		BEG E	GSB ①	180.00
	For Field Angle Traverse			
3	Key in reference azimuth away from beginning point.	REF AZ (D.MS)	R/S	Az (D.MS)
4	Key in field angle:			
	Angle right	ang. right	GSB ②	Az (D.MS)
	or Angle left (−)	−ang. left	GSB ②	Az (D.MS)
	or Deflection right	deflect. right	GSB ③	Az (D.MS)
	or Deflection left (−)	−deflect. left	GSB ③	Az (D.MS)
5	Key in horizontal distance and compute coordinates	HD	R/S	N
			R/S	E
	or			
	For Bearing Traverse			
3′	Key in bearing and quadrant code.	BRG (D.MS)	ENTER♦	
		QD	GSB ④	Az (D.MS)
4′	Key in horizontal distance and compute coordinates.	HD	R/S	N
			R/S	E
	Repeat steps 3, 4, 5, or 3′, 4′ for successive courses.			
6	For closed figure: Compute area, error distance, and error azimuth		GSB ⑤	Area
			R/S	Error Dist.
			R/S	Error Az (D.MS)

Example 1:

Field Angle Traverse

Traverse the figure below starting at $\frac{\text{N } 150}{\text{E } 400}$.

Keystrokes: ### Outputs:

Keystrokes	Outputs	
150 ENTER↑ 400 GSB 1 ⟶	180.0000	
311.3955 R/S ⟶	131.3955	
113.3455 GSB 2 ⟶	65.1450	
177.966 R/S ⟶	224.5150	(N)
R/S ⟶	561.6150	(E)
100.2455 CHS GSB 3 ⟶	324.4955	
161.491 R/S ⟶	356.5285	(N)
R/S ⟶	468.6000	(E)
87.3559 GSB 2 ⟶	232.2554	
203.690 R/S ⟶	232.3372	(N)
R/S ⟶	307.1498	(E)
100.4559 CHS GSB 3 ⟶	131.3955	
124 R/S ⟶	149.9048	(N)
R/S ⟶	399.7829	(E)
GSB 5 ⟶	26558.8204	(Area)
R/S ⟶	0.2371	(Error distance)
R/S ⟶	246.1844	(Error azimuth)

Example 2:

Bearing Traverse

Traverse the figure below starting at $\dfrac{\text{N }100}{\text{E }500}$.

Keystrokes:
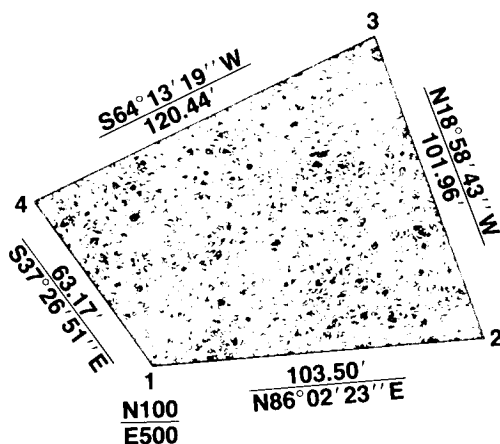
Keystrokes	Outputs	
100 ENTER↑ 500 GSB ① ⟶	180.0000	
86.0223 ENTER↑ 1 GSB ④ ⟶	86.0223	
103.50 R/S ⟶	107.1482	(N)
R/S ⟶	603.2529	(E)
18.5843 ENTER↑ 4 GSB ④ ⟶	341.0117	
101.96 R/S ⟶	203.5657	(N)
R/S ⟶	570.0939	(E)
64.1319 ENTER↑ 3 GSB ④ ⟶	244.1319	
120.44 R/S ⟶	151.1880	(N)
R/S ⟶	461.6395	(E)
37.2651 ENTER↑ 2 GSB ④ ⟶	142.3309	
63.17 R/S ⟶	101.0366	(N)
R/S ⟶	500.0490	(E)
GSB ⑤ ⟶	8855.4922	(Area)
R/S ⟶	1.0378	(Error distance)
R/S ⟶	2.4219	(Error azimuth)

Remarks:

- If the user does not desire to do Field Angle Traverse, steps 012 through 026 may be eliminated; if he does not desire to do Bearing Traverse, steps 064 through 080 may be eliminated.

- Angles left and deflections left must be entered as negative numbers.

- This program assumes the calculator is set in DEG mode.
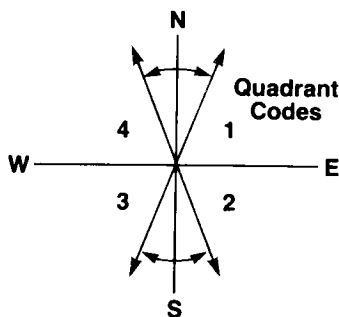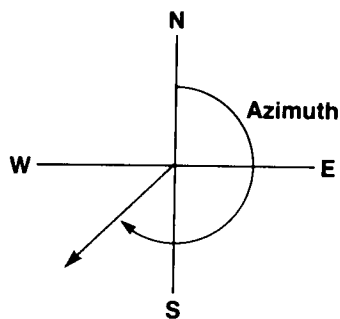
HORIZONTAL CURVE LAYOUT

This program calculates various field data for layout of a horizontal circular curve. The required information on the curve is the PC station and the radius or degree of curve. With this data one computes successively the arc length, deflection angle from tangent to long chord, the long chord from PC to current station, and the short chord from previous station to current station. In addition, the tangent offset and tangent distance are available if desired.

If the central angle is known the program also will compute the total arc length from PC to PT, the station PT and the length of the tangent from PC to PI.

In the program, stations are entered in the form XXXX.XX for station XX+XX.XX. For example: 20 + 10.00 is entered as 2010.00. The degree of curve D, (or central angle subtending an arc of 100 ft.) is entered in degrees with a negative sign, *always*.

PC Deflections
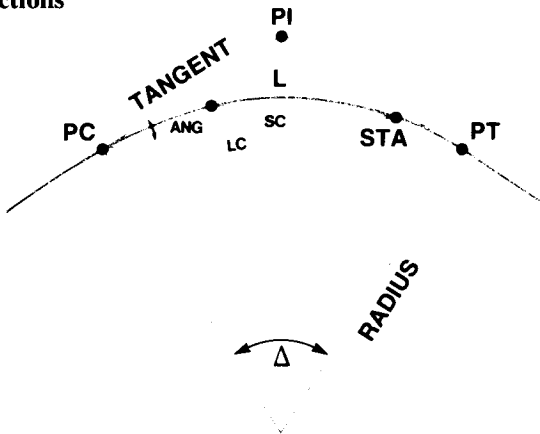
Field data output for PC deflections consist of:

STA-current station

ANG-deflection angle from tangent to long chord

LC-long chord from PC to current station

SC-short chord from previous station to current station

Δ-central angle

PI-point of intersection of tangents

PC, PT-ends of curve

L-Arc length

R-radius

Tangent Offsets and Distances
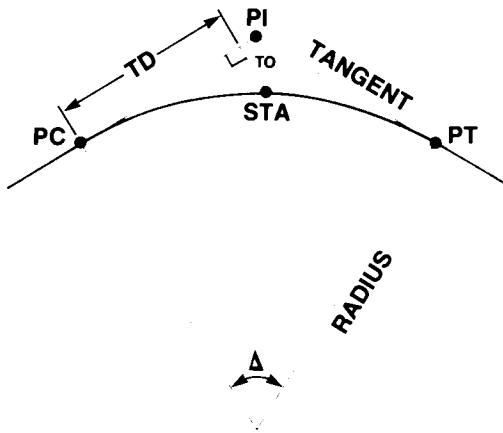
Field data output for tangent offsets consists of:

    STA-current station

    TD-tangent distance

    TO-tangent offset

    T-distance from PC to PI

01 *LBL1		50 SIN	Calculate TO
02 CLRG		51 x	
03 FIX4	Store R & D	52 STO8	
04 X≷C?		53 RCL5	Calculate TD
05 GSB0		54 RCL7	
06 STO1		55 COS	
07 P≷		56 x	
08 x		57 STO9	dsp LC
09 9		58 RCL5	
10 0		59 R/S	***
11 ÷		60 RCL4	– – – – – – – – –
12 STO2	Input PC	61 RC.2	
13 R↓		62 -	
14 STO3		63 GSB9	
15 STO4		64 x	
16 RTN		65 SIN	
17 *LBL0		66 RCL1	
18 CHS		67 2	
19 →H	Calculate R from D	68 x	
20 P≷		69 x	*** Calculate SC
21 ...		70 RTN	– – – – – – – – –
22 1/x		71 *LBL9	
23 1		72 9	
24 8		73 0	$\dfrac{90}{\pi R}$
25 EEX		74 P≷	
26 3		75 ÷	
27 ...		76 RCL1	Input Δ
28 RTN		77 ÷	
29 *LBL2	Input station	78 RTN	– – – – – – – – –
30 RCL4		79 *LBL3	
31 ST.2		80 →H	
32 R↓		81 2	
33 STO4		82 ÷	
34 RCL3		83 STO6	
35 -		84 GSB9	
36 R/S	*** Calculate L	85 ÷	Calculate L
37 GSB?	– – – – – – – – –	86 R/S	***
38 ...		87 RCL3	
39 STO7		88 +	Calculate PT
40 →HMS		89 R/S	*** – – – – – – –
41 R/S	*** Calculate ANG	90 RCL6	
42 RCL7	– – – – – – – – –	91 TAN	
43 SIN		92 RCL1	
44 RCL1		93 x	***
45 x		94 R/S	Calculate T
46 2			
47 x	Calculate LC		
48 STO5			
49 RCL7			

REGISTERS

0	1 R	2 Ft/Deflect	3 PC	4 STA Current	5 LC
6 Δ/2	7 ANG	8 TO	9 TD	.0	.1
.2 Prev. Sta.	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input beginning station of curve	PC	ENTER♦	PC
3	Input radius	R	GSB 1	
	or degree of curve (as a			
	negative number)	–D (D.MS)	GSB 1	
3′	Radius or degree of curve are		RCL 1	R
	available if desired.		RCL 2	D
4	Input station	STA	GSB 2	L (Arc. length)
			R/S	def. angle
			R/S	long chord
			R/S	short chord
4′	Tangent offset, TO, and tangent			
	distance, TD. are available if			
	desired.		RCL 8	TO
			RCL 9	TD
5	Input central angle	Δ (D.MS)	GSB 3	Arc. length
			R/S	station PT
			R/S	T, length of tan.

Example:

Compute field data for a curve with a central angle of 35° 30′ and degree of curve of 12° 30′. The PC station is 7 + 85.40.

Keystrokes:		Outputs:	
785.40 ENTER♦ 12.30			
CHS GSB 1 ——————→		785.40	(PC)
RCL 1 (if desired)——————→		458.3662	(R)
For Station 8:			
800 GSB 2 ——————→		14.6000	(L)
R/S ——————————→		.5445	(ANG)
R/S ——————————→		14.5994	(LC)
R/S ——————————→		14.5994	(SC)
RCL 8 (if desired)——————→		.2325	(TO)
RCL 9 (if desired)——————→		14.5975	(TD)

For Station 9:

900 GSB 2 ⟶	114.6000	(L)
R/S ⟶	7.0945	(ANG)
R/S ⟶	114.3018	(LC)
R/S ⟶	99.8018	(SC)
RCL 8 ⟶	14.2516	(TO)
RCL 9 ⟶	113.4098	(TD)

For Station 10:

1000 GSB 2 ⟶	214.6000	(L)
R/S ⟶	13.2445	(ANG)
R/S ⟶	212.6454	(LC)
R/S ⟶	99.8018	(SC)
RCL 8 ⟶	49.3252	(TO)
RCL 9 ⟶	206.8455	(TD)
35.30 GSB 3 ⟶	284.0000	(L)
R/S ⟶	1069.4000	(PT)
R/S ⟶	146.7242	(T)

Now calculate field data for PT:

1069.40 GSB 2 ⟶	284.000	(L)
R/S ⟶	17.4500	(ANG)
R/S ⟶	279.4790	(LC)
R/S ⟶	69.3337	(SC)

COORDINATE TRANSLATION AND ROTATION

This program allows for two-dimensional translation and rotation of coordinate axes. Suppose the origin of a coordinate system is translated to a new point, (x_0, y_0), and the x and y axes are rotated through an angle α to give new axes, x' and y'. A point P having coordinates (x, y) with respect to the old system of x and y axes, now has coordinates (x', y') with respect to the new axes. Given α and one pair of coordinates, the program allows you to find the other pair of coordinates.
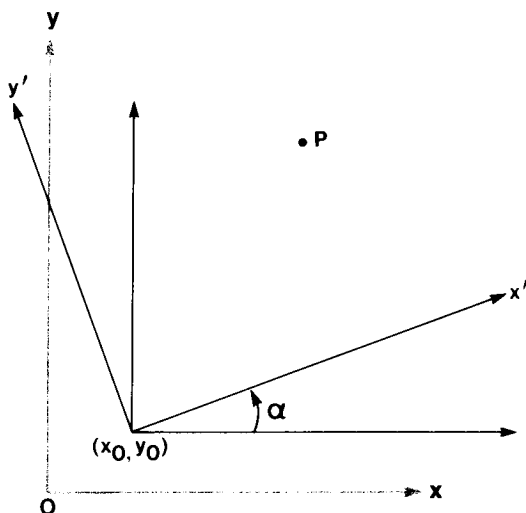
Equations:

Let Rect (r, θ) denote the operation ⟦→R⟧ when r is in the X-register and θ is in the Y-register. Let Pol (x, y) denote the operation ⟦→P⟧ when x is in X and y is in Y.

Then $(x', y') = $ Rect $(r, \theta - \alpha)$
    where $(r, \theta) = $ Pol $(x - x_0, y - y_0)$
and $(x, y) = (x_0, y_0) + $ Rect $(r', \theta' + \alpha)$
    where $(r', \theta') = $ Pol (x', y')

Remarks:

* The program may be used to solve a problem of translation only, or of rotation only, or of combined translation and rotation. If the problem involves translation alone, a value of $\alpha = 0$ must be input. For rotation alone, the values $x_0 = y_0 = 0$ must be input.
* The program assumes the following sign convention: α should be input as a positive number if the rotation is counterclockwise, and negative if clockwise.
* This program assumes the calculator is set in DEG mode.

01 *LBL0			
02 STO7	$x_0 \uparrow y_0$		
03 R↓	Stored		
04 STO9			
05 R↓			
06 STO8			
07 RTN	‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾		
08 *LBL1			
09 RCL9	Convert x, y to x′, y′		
10 −			
11 X⇄Y			
12 RCL8			
13 −			
14 →P			
15 X⇄Y			
16 RCL7			
17 −			
18 X⇄Y			
19 →R			
20 R/S	***		
21 X⇄Y	***		
22 RTN	‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾ ‾		
23 *LBL2			
24 X⇄Y	Convert x′, y′ to x, y		
25 →P			
26 X⇄Y			
27 RCL7			
28 +			
29 X⇄Y			
30 →R			
31 RCL8			
32 +			
33 R/S	***		
34 X⇄Y			
35 RCL9			
36 +	***		
37 RTN			

REGISTERS

0	1	2	3	4	5
6	7 θ	8 x_0	9 y_0	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Initialize:			
	Key in new origin	x_0	**ENTER↑**	
		y_0	**ENTER↑**	
	Key in angle of rotation (observe			
	proper sign)	α	**GSB** **0**	x_0
3	Key in old coordinates and			
	calculate coordinates in new			
	system	x	**ENTER↑**	
		y	**GSB** **1**	x'
			R/S	y'
4	Key in new coordinates and			
	calculate coordinates in old			
	system	x'	**ENTER↑**	
		y'	**GSB** **2**	x
			R/S	y

Example 1:

The origin of a coordinate system is translated to the point $(-1, 4)$ and rotated $30°$ in a positive (counterclockwise) direction. Find the new coordinates of the point whose coordinates in the old system are $(1, 3)$. If the coordinates of a point in the new system are $(5, 7)$, what are its coordinates in the old system?

Keystrokes: **Outputs:**

Keystrokes		Outputs	
1 **CHS** **ENTER↑** 4 **ENTER↑**			
30 **GSB** **0** ⟶		−1.00	
1 **ENTER↑** 3 **GSB** **1** ⟶		1.23	(x')
R/S ⟶		−1.87	(y')
5 **ENTER↑** 7 **GSB** **2** ⟶		−0.17	(x)
R/S ⟶		12.56	(y)

Example 2:

Backpacker Will B. Bushed's route will take him cross-country from the marked trails of an area. He knows that he will have to check his compass frequently against his map over this terrain, and regrets that the map is in such an inconvenient format for his purposes. In the first place, the grid lines on his map represent distances in feet from an origin about 25 miles away, resulting in such large numbers that the calculations are difficult. Secondly, the map's grid is based on true north while his compass readings are relative to magnetic north, a variation of 17°.

Before he leaves home, Bushed decides to draw a rough version of the map for his own convenience, locating his origin at the grid point (54 000, 118 000) and rotating his axes by 17° in a clockwise direction. As a first step, he wants to find the new coordinates of the bridge and the peak of the hill, whose coordinates in the old system are (55 750, 119 300) and (57 450, 120 500) respectively.
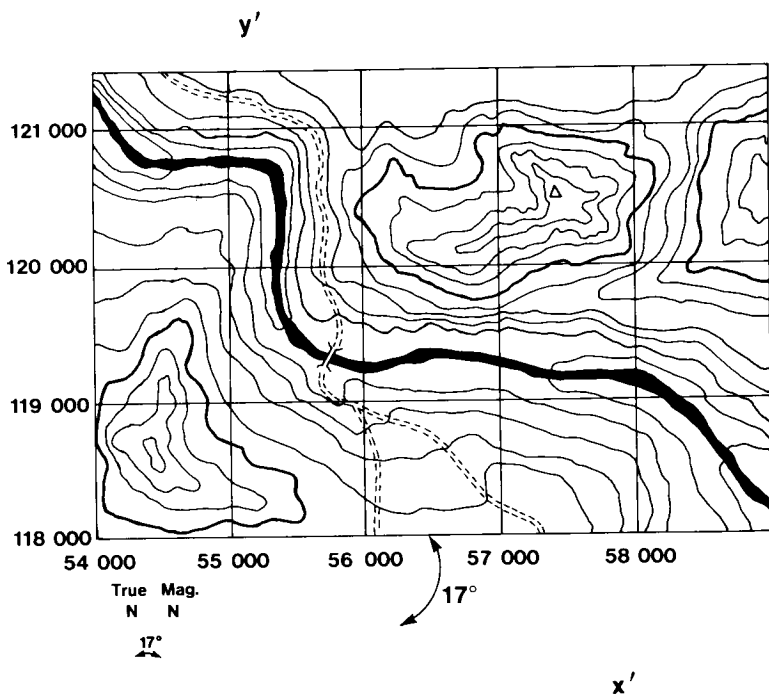
Keystrokes: **Outputs:**

54000 ENTER↑ 118000 ENTER↑
17 CHS GSB 0 55750
ENTER↑ 119300 GSB 1 ──────────→   1293.45
R/S ──────────────────────→   1754.85

The new coordinates of the bridge are (1293, 1755).

57450 ENTER↑ 120500
GSB 1 ──────────────────→   2568.32
R/S ──────────────────────→   3399.44
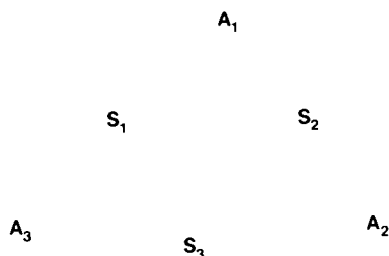
The new coordinates of the peak are (2568, 3399).

TRIANGLE SOLUTIONS

This program may be used to find the sides, the angles, and the area of a plane triangle.

A_1

S_1 S_2

A_3 A_2

S_3

In general, the specification of any three of the six parameters of a triangle (3 sides, 3 angles) is sufficient to define the triangle. (The exception is that three angles will not define a triangle.) There are thus five possible cases that this program will handle: two sides and the included angle (SAS), two angles and the included side (ASA), two sides and the adjacent angle (SSA—an ambiguous case), two angles and the adjacent side (AAS), and three sides (SSS).

The results are stored in storage registers 0 through 6 as follows:

AREA	Register 0
SIDE 1	Register 1
ANGLE 1	Register 2
SIDE 2	Register 3
ANGLE 2	Register 4
SIDE 3	Register 5
ANGLE 3	Register 6

Equations:

SAS (S_1, A_1, S_2):

$$S_3 = \sqrt{S_1{}^2 + S_2{}^2 - 2 S_1 S_2 \cos A_1}$$

$$A_2 = \tan^{-1} \frac{S_1 \sin A_1}{S_2 - S_1 \cos A_1}$$

$$A_3 = \cos^{-1} \left[- \cos (A_1 + A_2) \right]$$

ASA (A_3, S_1, A_1):

$$S_2 = S_1 \frac{\sin A_3}{\sin A_2} = S_1 \frac{\sin A_3}{\sin (A_1 + A_3)}$$

Now go to SAS.

SSA (S$_1$, S$_2$, A$_2$):

$$A_3 = \sin^{-1}\left(\frac{S_2 \sin A_2}{S_1}\right)$$

$$A_1 = \cos^{-1}\left[-\cos(A_2 + A_3)\right]$$

Now go to SAS.

AAS (A$_2$, A$_1$, S$_1$):

$$S_2 = S_1\frac{\sin A_3}{\sin A_2} = S_1\frac{\sin(A_1 + A_2)}{\sin A_2}$$

Now go to SAS.

SSS (S$_1$, S$_2$, S$_3$):

$$A_1 = \cos^{-1}\left(\frac{S_1^2 + S_2^2 - S_3^2}{2\,S_1\,S_2}\right)$$

Now go to SAS.

$$\text{AREA:} = \tfrac{1}{2}\,S_1\,S_3\,\sin A_3.$$

Remarks:

Any angular mode may be used.

Note that the triangle described by the program does not conform to standard triangle notation; i.e., A$_1$ is not opposite S$_1$.

Angles must be entered as decimals. The ▸H conversion can be used to convert degrees, minutes, and seconds to decimal degrees.

Accuracy of solution may degenerate for triangles containing extremely small angles.

01 *LBL1		
02 RCL2	SAS, Calculate A_2, S_3 by	
03 RCL1	law of cosines	
04 →R		
05 RCL3		
06 X≷Y		
07 -		
08 →P		
09 STO5		
10 X≷Y		
11 STO4		
12 RCL2		
13 +		
14 GSB0		
15 STO6	---------------	
16 SIN		
17 ×		
18 RCL1	Area	
19 ×		
20 2		
21 ÷		
22 STO0	***	
23 RTN	---------------	
24 *LBL2		
25 RCL1		
26 RCL3	SSS, Find A_1 by law of	
27 →P	cosines then go to SAS	
28 X²		
29 RCL5		
30 X²		
31 -		
32 RCL1		
33 RCL3		
34 ×		
35 2		
36 ×		
37 ÷		
38 COS⁻¹		
39 STO2		
40 GTO1	---------------	
41 *LBL3		
42 RCL6		
43 SIN		
44 RCL2	ASA, Find S_2 then go to	
45 RCL6	SAS	
46 +		
47 SIN		
48 ÷		
49 RCL1		

50 ×		
51 STO3		
52 GTO1		
53 *LBL4	---------------	
54 RCL3		
55 RCL4	SSA, Find A_1, go to SAS	
56 SIN	(Solution #1)	
57 RCL1		
58 ÷		
59 ×		
60 SIN⁻¹		
61 RCL4		
62 +		
63 GSB0		
64 STO2		
65 GSB1		
66 RCL1	Two solution exist if	
67 RCL3	$S_2 > S_1$	
68 X≤Y?		
69 RTN		
70 R/S	Solve for Solution #2	
71 RCL6		
72 GSB0		
73 STO6		
74 RCL4		
75 +		
76 GSB0		
77 STO2		
78 GTO1		
79 *LBL5	---------------	
80 RCL4		
81 RCL2	AAS, Find S_2 go to SAS	
82 +		
83 SIN		
84 RCL4		
85 SIN		
86 ÷		
87 RCL1		
88 ×		
89 STO3		
90 GTO1	---------------	
91 *LBL8		
92 COS		
93 CHS		
94 COS⁻¹		
95 RTN		

REGISTERS

| 0 Area	1 Side 1	2 Angle 1	3 Side 2	4 Angle 2	5 Side 3
6 Angle 3	7	8	9	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Select case and key in data:			
	2A: SAS (2 sides & included			
	angle)			
	Side 1	S_1	STO 1	
	Angle 1	A_1	STO 2	
	Side 2	S_2	STO 3	
			GSB 1	output
	2B: SSS (3 sides)			
	Side 1	S_1	STO 1	
	Side 2	S_2	STO 3	
	Side 3	S_3	STO 5	
			GSB 2	output
	2C: ASA (2 angles & included			
	side)			
	Angle 3	A_3	STO 6	
	Side 1	S_1	STO 1	
	Angle 1	A_1	STO 2	
			GSB 3	output
	2D: SSA (2 sides & adjacent			
	angle) Side 1 (opposite side)	S_1	STO 1	
	Side 2 (adjacent side)	S_2	STO 3	
	Angle 2 (adjacent angle)	A_2	STO 4	
			GSB 4	solution # 1*
			R/S	solution # 2**
	2E: AAS (2 angles & adjacent			(If it exists)
	side)			
	Angle 1 (adjacent angle)	A_1	STO 2	
	Angle 2 (opposite angle)	A_2	STO 4	
	Side 1 (adjacent side)	S_1	STO 1	
			GSB 5	output

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
3	Obtain solution by reviewing registers (use print reg. command if applicable).			
	recall reg. 0		`RCL` `0`	area
	recall reg. 1		`RCL` `1`	S_1
	recall reg. 2		`RCL` `2`	A_1
	recall reg. 3		`RCL` `3`	S_2
	recall reg. 4		`RCL` `4`	A_2
	recall reg. 5		`RCL` `5`	S_3
	recall reg. 6		`RCL` `6`	A_3
	* Review registers at this point for solution #1. ** Press `R/S`, *once only,* for solution #2, (pressing `R/S` more than once will give erroneous results.)			

Example 1:

Find the angles and the area for the following triangle.

2.0   1.0

2.75
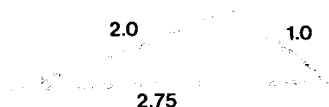
Keystrokes: **Outputs:**

2 `STO` `1` 1 `STO` `3`
2.75 `STO` `5` `GSB` `2`
(and recall registers) ⟶   Reg 0   0.77   (Area)

Reg 1	2.00	(S_1)
Reg 2	129.84	(A_1)
Reg 3	1.00	(S_2)
Reg 4	33.95	(A_2)
Reg 5	2.75	(S_3)
Reg 6	16.21	(A_3)

Example 2:

A surveyor is to find the area and dimensions of a triangular land parcel. From point A, the distances to B and C are measured with an electronic distance meter. The angle between AB and AC is also measured. Find the area and other dimensions of the triangle.

A

171.63
meters   297.35
                meters
          98° 12′

B   **C**
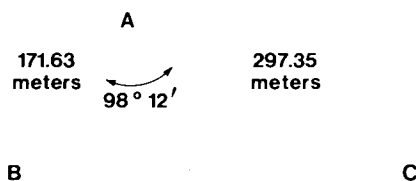
This is a side-angle-side problem where:

$$S_1 = 171.63, \ A_1 = 98°12' \text{ and } S_2 = 297.35.$$

Keystrokes: **Outputs:**

171.63 STO 1 98.12
9 →H STO 2 297.35
STO 3 GSB 1
(and recall registers) ⟶ Reg 0   25256.21 (Area (m²))
                          Reg 1   171.63 (AB, m)
                          Reg 2   98.20 (ANG. A)
                          Reg 3   297.35 (AC, m)
                          Reg 4   27.83 (ANG. C)
                          Reg 5   363.91 (CB, m)
                          Reg 6   53.97 (ANG. B)

Example 3:

A pilot wishes to fly due north. The wind is reported as 25 knots at 77°. Because winds are reported opposite to the direction they blow, this is interpreted as 77 + 180 or 257°. The true airspeed of the aircraft is 140 knots. What heading (HDG) should be flown? What is the ground speed (GS)?
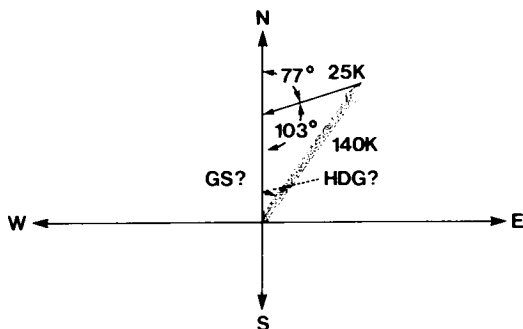
By subtracting the wind direction from 180 (yielding an angle of 103°), the problem reduces to a S_1, S_2, A_2 triangle.

Keystrokes: **Outputs:**

140 STO [1] 25 STO [3]
103 STO [4] GSB [4]
(and recall registers) ⟶   25.00   (Side 2)

Reg 0	1610.64	
Reg 1	140.00	(TAS)
Reg 2	66.98	
Reg 3	25.00	(WIND VEL.)
Reg 4	103.00	
Reg 5	132.24	(GS)
Reg 6	10.02	(HDG)

[R/S] ⟶ No Operation
(No Second Solution)

Thus, the pilot should fly a heading 10.02° east due north. His ground speed equals 132.24 knots.

Example 4:

Two angles and an adjacent side of a triangle are known. Calculate the area of the triangle, the other two sides and the third angle. The known side is 19.6 ft. and the angle adjacent is 61.06°. The opposite angle is 40.25°.

This is an AAS case where S_1 = 19.6 ft., A_1 = 61.06° and A_2 = 40.25°.

Keystrokes: **Outputs:**

19.6 STO 1 61.06
STO 2 40.25 STO 4
GSB 5 (and recall

registers) ⟶   Reg 0   255.11   (Area (ft²))
                      Reg 1   19.60   (S_1, ft)
                      Reg 2   61.06   (A_1, deg)
                      Reg 3   29.75   (S_2, ft)
                      Reg 4   40.25   (A_2, deg)
                      Reg 5   26.55   (S_3, ft)
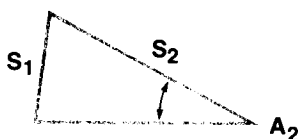   Reg 6   78.69   (A_3, deg)

Example 5:

Given 2 sides and a nonincluded angle solve for the triangle:

Side 1 = 25.6
Side 2 = 32.8
Angle 2 = 42.3°

(Note: Since $S_1 < S_2$ and $A_2 < 90°$, 2 solutions exist.)

Keystrokes: **Outputs:**

25.6 STO 1 32.8 STO
3 42.3 STO 4 GSB 4
(and recall registers) ⟶   Reg 0   410.85   (Area)
                            Reg 1   25.60   (S_1)
                            Reg 2   78.12   (A_1)
(Solution #1)   Reg 3   32.80   (S_2)
                            Reg 4   42.30   (A_2)
                            Reg 5   37.22   (S_3)
                            Reg 6   59.58   (A_3)

R/S (and recall
registers) ⟶   Reg 0   124.68   (Area)
                      Reg 1   25.60   (S_1)
                      Reg 2   17.28   (A_1)
                      Reg 3   32.80   (S_2)
(Solution #2)   Reg 4   42.30   (A_2)
                      Reg 5   11.30   (S_3)
                      Reg 6   120.42   (A_3)

Example 6:

A triangle has angles of 64°32′ and 35°06′ with the included side 20.96 feet long. Solve for the remainder of the triangle.

Keystrokes: **Outputs:**

64.32 **g** **←H** **STO** **6**
20.96 **STO** **1** 35.06 **g** **←H**
STO **2** **GSB** **3** (and
recall registers) ⟶ Reg 0   115.66   (Area (ft^2))

Reg 1	20.96	(S_1, ft)
Reg 2	35.10	(A_1)
Reg 3	19.19	(S_2, ft)
Reg 4	80.37	(A_2)
Reg 5	12.22	(S_3, ft)
Reg 6	64.53	(A_3)

CIRCLE DETERMINED BY THREE POINTS

This program calculates the center (x_0, y_0) and radius (r) of a circle given three non-collinear points.

Equations:

Circle determined by three points:

$$y_0 = \frac{K_2 - K_1}{N_2 - N_1} \cdot x_0 = K_2 - N_2 y_0$$

$$r = \sqrt{(x_3 - x_0)^2 + (y_3 - y_0)^2}$$

where:

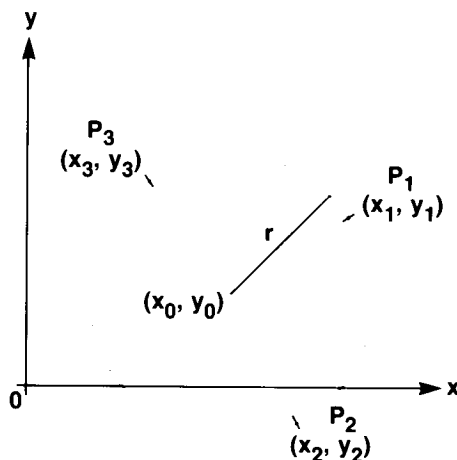$$K_1 = \frac{(x_2 - x_1)(x_2 + x_1) + (y_2 - y_1)(y_2 + y_1)}{2(x_2 - x_1)}$$

$$K_2 = \frac{(x_3 - x_1)(x_3 + x_1) + (y_3 - y_1)(y_3 + y_1)}{2(x_3 - x_1)}$$

$$N_1 = \frac{y_2 - y_1}{x_2 - x_1}$$

$$N_2 = \frac{y_3 - y_1}{x_3 - x_1}$$

Remarks:

If $x_1 = x_2$ or $x_1 = x_3$ in the calculation of the center and radius of a circle, then point 1 replaces point 3, point 3 replaces point 2 and point 2 replaces point 1.

01 *LBL1		50 RTN	***
02 RCL3		51 *LBL6	
03 RCL1		52 RCL1	
04 X=Y?		53 RCL2	
05 GTO9		54 →P	
06 RCL5	Check for $x_1 = x_2$ or	55 X²	
07 X=Y?	$x_1 = x_3$	56 STO0	
08 GTO8		57 RCL3	
09 *LBL5		58 RCL4	
10 GSB6		59 →P	
11 ST.3		60 X²	
12 X≷Y		61 RCL0	
13 STO7		62 -	Subroutine to
14 RCL3		63 RCL3	
15 RCL5		64 RCL1	Calculate K_1, K_2,
16 STO3		65 -	
17 X≷Y		66 2	N_1 and N_2
18 STO5		67 >	
19 RCL4	Calculate y_0, x_0, or r	68 ÷	
20 RCL6		69 RCL4	
21 STO4		70 RCL2	
22 X≷Y		71 -	
23 STO6		72 RCL3	
24 GSB6		73 RCL1	
25 ST.4		74 -	
26 X≷Y		75 ÷	
27 STO8		76 RTN	
28 RCL7		77 *LBL9	
29 -		78 RCL5	
30 RC.4		79 X≷Y	
31 RC.3		80 GTO8	Check for $x_1 = x_2 = x_3$
32 -		81 *LBL8	
33 ÷		82 RCL1	
34 ST.2		83 RCL3	
35 RC.4		84 RCL5	
36 x		85 STO3	
37 RCL8		86 R↓	
38 -		87 STO1	
39 CHS		88 R↓	
40 ST.1		89 STO5	Swap for $x_1 = x_2$ or
41 R/S	***	90 RCL2	$x_1 = x_3$
42 RC.2		91 RCL4	
43 R/S	***	92 RCL6	
44 RCL4		93 STO4	
45 -		94 R↓	
46 RCL3		95 STO2	
47 RC.1		96 R↓	
48 -		97 STO6	
49 →P		98 GTO5	

REGISTERS					
0 Used	1 x_1	2 y_1	3 x_2	4 y_2	5 x_3
6 y_3	7 k_1	8 k_2	9 r	.0	.1 x_0
.2 y_0	3 N_1	.4 N_2	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input x_1, y_1; x_2, y_2; x_3, y_3	x_1	**STO** 1	x_1
		y_1	**STO** 2	y_1
		x_2	**STO** 3	x_2
		y_2	**STO** 4	y_2
		x_3	**STO** 5	x_3
		y_3	**STO** 6	y_3
3	Calculate x_0, y_0, and r		**GSB** 1	x_0
			R/S	y_0
			R/S	r
4	For a new case go to step 2.			

Example:

What circle contains the points $(1, 1)$, $(3.5, -7.6)$ and $(12, 0.8)$?

Keystrokes: Outputs:

1 **STO** 1 1 **STO** 2 3.5
STO 3 7.6 **CHS** **STO** 4 12
STO 5 0.8 **STO** 6
GSB 1 ⟶ 6.45 (x_0)
R/S ⟶ −2.08 (y_0)
R/S ⟶ 6.26 (r)

INTERSECTIONS OF LINES AND LINES, LINES AND CIRCLES, AND CIRCLES AND CIRCLES

This program calculates the point of intersection of two lines, the points of intersection of a coplanar circle and line, or the points of intersection of two coplanar circles.

There are three sub-programs, i.e.,

1. Calculates intersections of lines and lines.

    Lines may be specified by two points (x_1, y_1, and x_2, y_2), or by one point and an angle (θ), where θ is the angle from the positive x-axis to the line.

2. Calculates intersections of circles and lines.

    Lines are specified by two points (x_1, y_1, and x_2, y_2).

    Circles are specified by their center coordinates (x_0, y_0) and the radius (r).

3. Calculates intersections of circles and circles.

    Circles are specified by their center coordinates (x_0, y_0) and the radius (r).

Equations:

Line-Line Intersection:

$$x_p = \frac{x_1 \tan \theta_1 - x_2 \tan \theta_2 + y_2 - y_1}{\tan \theta_1 - \tan \theta_2}$$

$$y_p = y_1 + (x - x_1) \tan \theta_1$$

$$y_1 = x_1 \tan \theta_1 + C_1$$

$$y_2 = x_2 \tan \theta_2 + C_2$$
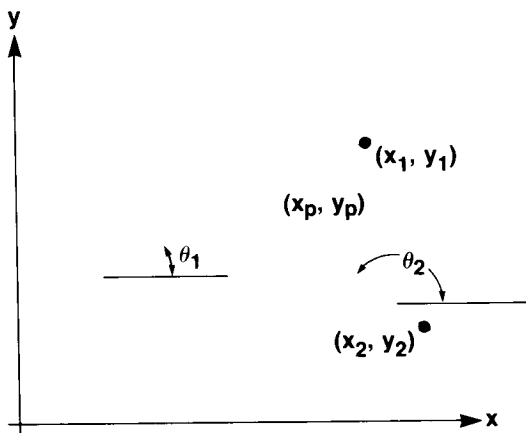
Circle-Line Intersections:

$$x_{p1} = x_1 + P_1\cos\theta$$
$$y_{p1} = y_1 + P_1\sin\theta$$
$$x_{p2} = x_1 + P_2\cos\theta$$
$$y_{p2} = y_1 + P_2\sin\theta$$

where:

P_1 and P_2 are the roots of

$$P^2 - 2 D \cos(\theta - \alpha) P + D^2 - r^2 = 0$$

$$\theta = \tan^{-1}\left[\frac{y_2 - y_1}{x_2 - x_1}\right]$$

$$\alpha = \tan^{-1}\left[\frac{y_0 - y_1}{x_0 - x_1}\right]$$

$$D = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$
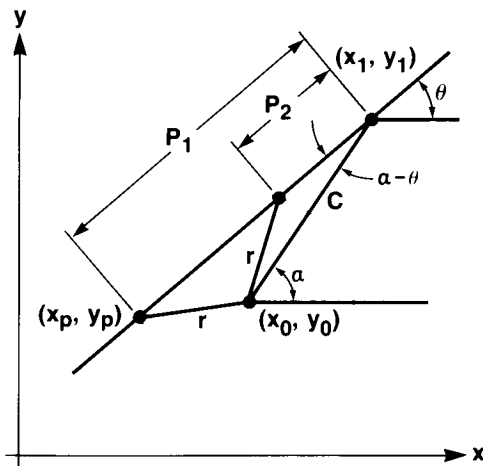
Circle-Circle Intersections:

$$x_{p1} = x_{01} + r_1 \cos(\theta + \alpha)$$
$$y_{p1} = y_{01} + r_1 \sin(\theta + \alpha)$$
$$x_{p2} = x_{01} + r_1 \cos(\theta - \alpha)$$

$$y_{p2} = y_{01} + r_1 \sin (\theta - \alpha)$$

$$\theta = \tan^{-1} \left(\frac{y_{02} - y_{01}}{x_{02} - x_{01}} \right)$$
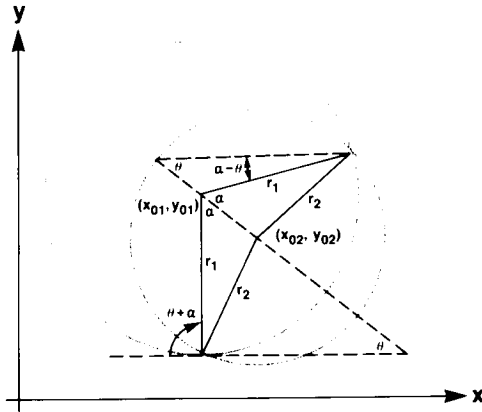
$$\alpha = \cos^{-1} \left[\frac{D^2 + r_1{}^2 - r_2{}^2}{2Dr_1} \right]$$

$$D = \sqrt{(x_{02} - x_{01})^2 + (y_{02} - y_{01})^2}$$

Intersections Part 1: Line-Line

01 *LBL3		50 -		
02 ST03		51 →P		
03 R+	Input one point and angle	52 R↓		
04 ST02	for the 1st point	53 RTN		
05 R↓		54 *LBL5		
06 STC1		55 RCL7		
07 GT05		56 RCL3		
08 *LBL4		57 X≷Y?		
09 ST07		58 GT00	To calculate x_p, y_p
10 R↓		59 ABS	
11 ST06	Input one point and angle	60 RC.0	
12 R+	for the 2nd point	61 X≠Y?	
13 ST05		62 GT06	
14 GT08		63 RCL1	
15 *LBL1		64 R/S	
16 ST02		65 RCL7	
17 X≷Y		66 TAN	
18 ST01		67 ×	
19 GSB6		68 RCL8	
20 ST03	Input coordinates of the	69 +	
21 *LBL9	1st point	70 RTN	
22 RCL2		71 *LBL6	
23 RCL1		72 RCL7	
24 RCL3		73 ABS	
25 TAN		74 RC.0	
26 ×		75 X≠Y?	
27 -		76 GT07	
28 ST04		77 RCL5	The 1st line is vertical.
29 RTN		78 R/S	
30 *LBL2		79 GT05	
31 ST06		80 *LBL7	
32 X≷Y		81 RCL8	
33 ST05		82 RCL4	
34 GSB6	Input coordinates of the	83 -	
35 ST07	2nd point.	84 RCL3	
36 *LBL8		85 TAN	
37 RCL6		86 RCL7	
38 RCL5		87 TAN	
39 RCL7		88 -	The 2nd line is vertical.
40 TAN		89 ÷	
41 ×		90 R/S	
42 -		91 *LBL9	
43 ST08		92 RCL3	
44 RTN		93 TAN	
45 *LBL6	Subroutine to find the	94 ×	
46 R↓	slope and constant.	95 RCL4	
47 -		96 +	
48 X≷Y		97 RTN	
49 RCL0			

REGISTERS

0 temp x_2	1 x_1'	2 y_1'	3 θ_1	4 c_1	5 x_2'
6 y_2'	7 θ_2	8 c_2	9	.0 90	.1
.2	3	4	5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

Intersections Part 2: Circle Line

01 *LBL2		50 +	
02 STO2		51 JX	
03 X≷Y		52 STO9	•••
04 STO1		53 RTN	
05 R↓		54 *LBL5	– – – – – – – – –
06 -		55 *LBL3	
07 X≷Y	Input x_1, y_1, x_2, y_2, and	56 RCL9	
08 RCL0	calculate	57 RCL6	
09 -		58 -	Calculate x_p
10 →P		59 RC.1	
11 R↓		60 ÷	
12 9		61 R/S	•••
13 0		62 RCL3	
14 X≷Y		63 x	
15 X=Y?		64 RCL4	
16 GTO5		65 +	
17 TAN		66 *LBL9	•••
18 STO3		67 RTN	
19 RCL2		68 *LBL4	– – – – – – – – –
20 RCL1		69 ÷	
21 RCL3		70 CHS	Calculate y_p
22 x		71 STx9	
23 -		72 GTO5	– – – – – – – – –
24 STO4		73 *LBL3	
25 RCL6		74 RCL7	
26 -		75 X²	
27 ST.2		76 RCL1	
28 RCL3		77 R/S	•••
29 x		78 RCL5	
30 RCL5		79 -	
31 -		80 X²	Calculate x_p and y_p for
32 STO8		81 -	vertical line.
33 RC.2		82 JX	
34 X²		83 ST.3	
35 RCL5		84 *LBL8	
36 X²		85 RCL6	
37 +		86 +	
38 RCL7		87 RTN	
39 X²		88 *LBL4	
40 -		89 RCL1	
41 RCL3		90 R/S	•••
42 X²		91 RC.3	
43 1		92 CHS	
44 +		93 GTO8	
45 ST.1			
46 x			
47 CHS			
48 RCL8			
49 X²			

REGISTERS					
0 temp x_2	1 x_1'	2 y_1'	3 $\tan\theta_1$	4 c_1	5 x_0
6 y_0	7 r	8 α	9 β	.0	.1 $(1 + m)$
.2 $c - y_0$	.3 Used	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

••• indicates that "Print X" may be inserted or used to replace "R/S".

Intersections Part 3: Circle-Circle

01	*LBL1			50	GSB9	
02	STO3			51	*LBL8	
03	R↓			52	RCL0	
04	STO2	Input x_{01}, y_{01}, r_1		53	SIN	Calculate x_{p2}, y_{p2}
05	R↓			54	RCL3	
06	STO1			55	×	
07	RTN			56	RCL2	
08	*LBL2			57	+	•••
09	STO6			58	RTN	
10	R↓			59	*LBL9	
11	STO5	Input x_{02}, y_{02}, r_2		60	STO0	
12	R↓			61	COS	
13	STO4			62	RCL3	
14	RTN			63	×	
15	*LBL3			64	RCL1	
16	RCL5			65	+	
17	RCL2			66	R/S	•••
18	-			67	RTN	
19	RCL4					
20	RCL1					
21	-					
22	→P					
23	STO8	Calculate x_{p1}, y_{p1}				
24	X≷Y					
25	STO7					
26	RCL8					
27	X²					
28	RCL3					
29	X²					
30	+					
31	RCL6					
32	X²					
33	-					
34	RCL8					
35	2					
36	×					
37	RCL3					
38	×					
39	÷					
40	COS⁻¹					
41	STO9					
42	RCL7					
43	+					
44	GSB9					
45	GTO8					
46	*LBL4					
47	RCL7					
48	RCL9					
49	-					

REGISTERS					
0 $\theta \pm \alpha$	1 x_{01}	2 y_{01}	3 r_1	4 x_{02}	5 y_{02}
6 r_2	7 θ	8 D	9 α	.0	.1
.2	.3	.4	.5	16	17
18	19	20	21	22	23
24	25	26	27	28	29

*** indicates that "Print X" may be inserted or used to replace "R/S".

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the corresponding program.			
2	For lines and lines, go to step 3. For circles and lines, go to step 9. For circles and circles, go to step 14.			
3	**Lines and Lines**			
4	Initialize	90	STO · 0	90
5	Input the first line:			
	by two points:			
	x_1	x_1	ENTER↑	
	y_1	y_1	ENTER↑	
	x_2	x_2	STO 0	
	y_2	y_2	GSB 1	c_1
	or			
	by one point and the angle	x_1	ENTER↑	
		y_1	ENTER↑	
		θ	GSB 3	c_1
6	Input the 2nd line:			
	by two points:			
	x'_1	x'_1	ENTER↑	
	y'_1	y'_1	ENTER↑	
	x'_2	x'_2	STO 0	
	y'_2	y'_2	GSB 2	c_2
	or			
	by one point and the angle	x'_1	ENTER↑	
		y'_1	ENTER↑	
		θ'	GSB 4	c_2
7	Calculate intersection point		GSB 5	x_p
			R/S	y_p
8	For a new case go to step 5.			

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
9	**Circles and Lines**			
10	Input the circle center			
	x_0	x_0	`STO` `5`	x_0
	y_0	y_0	`STO` `6`	y_0
	radius r	r	`STO` `7`	r
11	Input the line by two points			
	x_1	x_1	`ENTER↑`	
	y_1	y_1	`ENTER↑`	
	x_2	x_2	`STO` `0`	
	y_2	y_2	`GSB` `2`	c_2
12	Calculate the intersection points			
	x_{p1}		`GSB` `3`	x_{p1}
	y_{p1}		`R/S`	y_{p1}
	x_{p2}		`GSB` `4`	x_{p2}
	y_{p2}		`R/S`	y_{p2}
13	For a new case, go to step 9.			
14	**Circles and Circles**			
15	Input circle one			
	x_{01}	x_{01}	`ENTER↑`	
	y_{01}	y_{01}	`ENTER↑`	
	r_1	r_1	`GSB` `1`	x_{01}
16	Input circle two			
	x_{02}	x_{02}	`ENTER↑`	
	y_{02}	y_{02}	`ENTER↑`	
	r_2	r_2	`GSB` `2`	x_{02}
17	Calculate intersections			
	x_{p1}		`GSB` `3`	x_{p1}
	y_{p1}		`R/S`	y_{p1}
	x_{p2}		`GSB` `4`	x_{p2}
	y_{p2}		`R/S`	y_{p2}
18	For a new case, go to step 14.			

Example 1:

Find the intersection of the vertical line specified by two points:

$$P_1 = (0, 0)$$
$$P'_1 = (0, 50)$$

And the oblique line specified by one point and an angle:

$$P_2 = (10, 20)$$
$$\theta = 45°$$

Keystrokes: **Outputs:**

(Key in the first program)

90 STO ⋅ 0 ───────────→	90.00	
0 ENTER↑ ENTER↑ STO 0 50		
GSB 1 ───────────→	9.9999999 +99	(Neglect)
10 ENTER↑ 20 ENTER↑ 45		
GSB 4 ───────────→	10.00	
GSB 5 ───────────→	0.00	(x_p)
R/S ───────────→	10.00	(y_p)

Example 2:

Find the points of intersection for a circle with center at (0, 0) and radius 50, and the line containing the points (20, 30) and (0, −10).

Keystrokes: **Outputs:**

(Key in the second program)

0 STO 5 STO 6 50 STO 7 →	50.00	
20 ENTER↑ 30 ENTER↑ 0 STO		
0 10 CHS GSB 2 ───────→	111.36	
GSB 3 ───────────→	26.27	(x_{p1})
R/S ───────────→	42.54	(y_{p1})
GSB 4 ───────────→	−18.27	(x_{p2})
R/S ───────────→	−46.54	(y_{p2})

Example 3:

Calculate the points of intersection for circles at (0, 0) radius 50 and (90, 30) radius 70.

Keystrokes: Outputs:

(Key in the third program)

Keystroke	Output	
0 ENTER↑ ENTER↑ 50 GSB 1 →	0.00	
90 ENTER↑ 30 ENTER↑ 70		
GSB 2 ──────────→	90.00	
GSB 3 ──────────→	21.64	(x_{p1})
R/S ──────────→	45.07	(y_{p1})
GSB 4 ──────────→	44.36	(x_{p2})
R/S ──────────→	−23.07	(y_{p2})

HEWLETT **hp** PACKARD

1000 N.E. Circle Blvd., Corvallis, OR 97330

For additional Sales and Service Information contact your local Hewlett-Packard
Sales Office or call 800/648-4711. (In Nevada call collect 702/323-2704.)