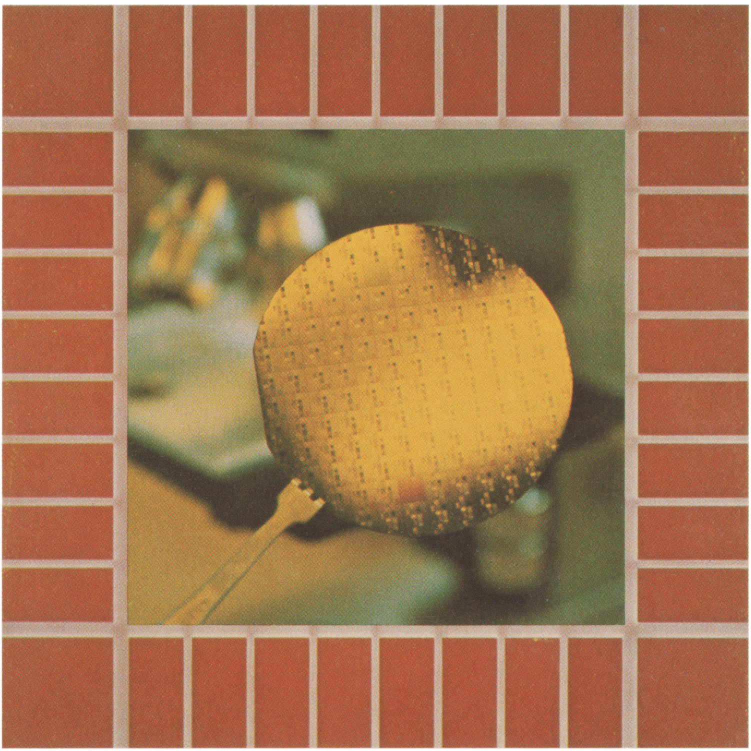
HEWLETT-PACKARD

HP-33E
APPLICATIONS



HEWLETT hp PACKARD

HP-33E

Applications

February, 1978

00033-90024

Printed in U.S.A.

Introduction

Congratulations on owning an HP-33E. We know you will be pleased with its quality, programmable power and ease of use. This application book is designed to help you get the best from your calculator, whether your interest is in solving problems or learning to get the most from the programming capabilities of the HP-33E.

The programs in this book have been chosen to provide answers to "real world" problems, to provide interesting programs and games and to illustrate programming techniques.

We are confident you will find this book useful and we welcome any comments or suggestions you may have.

Contents

A Word About Program Usage

Each program includes with it a brief description of the problem, the applicable equations, a listing of program keystrokes, a set of instructions for using the program and one or more example problems, including the actual keystrokes required for the solution.

Program listings are provided in the following format: (This example is from the Plotting/Graphing program, the first program in the book.)

KEY ENTRY	DISPLAY	KEY ENTRY	DISPLAY
[f] CLEAR [PRGM]	00	[CHS]	15- 32
[f] [→R]	01- 14 4	[RCL] 4	16- 24 4
[STO] 2	02- 23 2	[RCL] 3	17- 24 3
[x⇄y]	03- 21	[×]	18- 61
[STO] 3	04- 23 3	[+]	19- 51
0	05- 0	[RCL] 4	20- 24 4
[STO] 4	06- 23 4	[RCL] 2	21- 24 2
[RCL] 0	07- 24 0	[×]	22- 61

The leftmost column, headed KEY ENTRY, shows the keys which must be pressed to enter the program into program memory. All the key designations are identical with the way they appear on your keyboard. The second column, headed DISPLAY, shows the appearance of the display on the calculator as you key in the program. The first two numerals on the left are the line number, then a dash, then the numeric keycode corresponding to the keystrokes in the KEY ENTRY column. Storage register contents are shown at the end of the program listing.

The User Instructions form is your guide to using the program to solve your own problem. The first column, labeled STEP, gives the instruction step number. Steps are executed in sequential order except where otherwise noted. The INSTRUCTIONS column gives instructions and comments concerning the operations to be performed. The INPUT

DATA/UNITS column specifies the input data to be supplied, and the units of the data, if applicable. Data input keys consist of 0 thru 9 and decimal point (the numeric keys) [EEX] (enter exponent) and [CHS] (change sign). The KEYS column specifies the keys to be pressed after keying in the corresponding input data. The OUTPUT DATA/UNITS column specifies intermediate and final outputs and their units, where applicable.

The form is illustrated below for the same program, plotting/graphing.

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in program			
2	Store time interval	Δt	[STO] 0	
3	Store gravitational constant	g	[STO] 1	
4	Input angle and initial speed	θ	[ENTER↑]	
		v	[f] [PRGM]	
5	Perform steps 5 and 6 any			
	number of times: Display time		[R/S]	(t)
	and horizontal distance			x
6	Display height		[R/S]	y
7	To change θ or v, go to step 4.			
	To change Δt or g, go to			
	appropriate step, store new			
	value, then go to step 4.			

Step 1 requires you to key in the program. Switch the HP-33E to PRGM mode, depress [f] CLEAR [PRGM] and key in the program steps as shown on the complete listing.

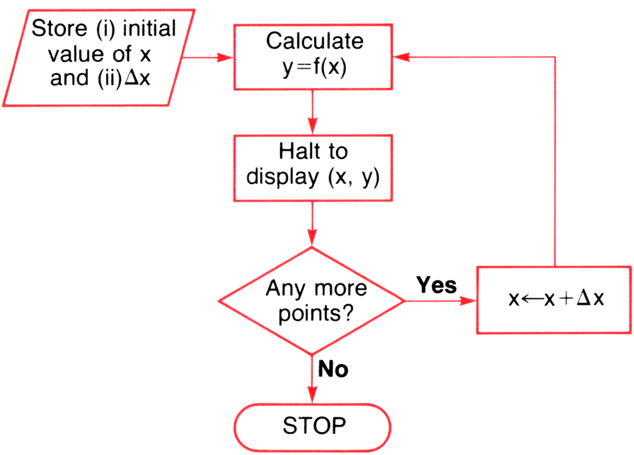
Steps 2 and 3 ask for storage of data. Switch the calculator to RUN mode and input and store the data as shown.

Mathematics

Plotting/Graphing

Most people who have labored through a ninth-grade algebra course probably still respond with a shudder to the word ''graph''. Evidently the tedium of finding $y = 3x^2 - 4x + 4$ for integer values of x from $-\infty$ to $+\infty$ has etched permanent memories in us all. Fortunately, we need not endure this tedium any longer. The HP-33E lends itself perfectly to this kind of repetitive calculation.

The basic idea is to generate (x, y) pairs by keying into program memory the keystrokes required to calculate y, assuming x is given. Then the user need only return to the top of memory, enter a value for x, press [R/S], and see y displayed within seconds. The process may be repeated for as many values of x as desired. The programmer can take this process one step further into automation by also having the calculator generate each new value of x, for example, by adding 1 to the old value, or, in general, by adding a specified increment Δx. A flowchart of the process is shown below.



The program used here to illustrate this process takes a slightly different tack. We will consider the problem of plotting the trajectory of a stone which is hurled into the air with an initial velocity v at an angle to the horizontal of θ. Neglecting drag due to friction with the atmosphere, the following equations describe the stone's x- and y-coordinates as functions of the time t:

$$x = vt\cos\theta \qquad\qquad y = vt\sin\theta - \frac{1}{2}gt^2$$

where: x = horizontal distance the stone has traveled

y = height of the stone

g = acceleration due to gravity
$\simeq 9.8\ m/s^2$
$\simeq 32\ ft/s^2$

These equations differ slightly from the usual graphing function in that y is not expressed directly as a function of x, but instead both x and y are expressed as functions of a third variable t. The points to be plotted are still the ordered pairs (x, y); but now it is the time t which should be incremented by an amount Δt.

Notes:

1. Any consistent set of units may be used.
2. This is *not* a general plotting/graphing program; it merely illustrates the method by application to a specific problem. However, some study of the program listing and the flowchart should enable the user to adapt the method to his own application.

Programming Remarks:

1. The components of the velocity in the horizontal and vertical directions, v_x and v_y, are computed in one step by a conversion of v and θ to rectangular coordinates ([f] [→R]). The values $v_x = v\cos\theta$ and $v_y = v\sin\theta$ are returned to the X- and Y-registers, respectively.
2. A pause ([f] [PAUSE]) is used in this program in a very typical manner, to display briefly the output variable t, whose values are simple (0.25, 0.50, 0.75, etc.) and do not need to be written down.

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
f →R	01– 14 4
STO 2	02– 23 2
x⇄y	03– 21
STO 3	04– 23 3
0	05– 0
STO 4	06– 23 4
RCL 0	07– 24 0
STO + 4	08– 23 51 4
RCL 4	09– 24 4
g x^2	10– 15 0
RCL 1	11– 24 1
×	12– 61
2	13– 2
÷	14– 71

KEY ENTRY	DISPLAY
CHS	15– 32
RCL 4	16– 24 4
RCL 3	17– 24 3
×	18– 61
+	19– 51
RCL 4	20– 24 4
RCL 2	21– 24 2
×	22– 61
RCL 4	23– 24 4
f PAUSE	24– 14 74
R↓	25– 22
R/S	26– 74
x⇄y	27– 21
R/S	28– 74
GTO 07	29– 13 07

REGISTERS			
R_0 Δt	R_1 g	R_2 v_x	R_3 v_y
R_4 t	R_5	R_6	R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in program			
2	Store time interval	Δt	[STO] 0	
3	Store gravitational constant	g	[STO] 1	
4	Input angle and initial speed	θ	[ENTER↑]	
		v	[f] [PRGM]	
5	Perform steps 5 and 6 any			
	number of times: Display time		[R/S]	(t)
	and horizontal distance			x
6	Display height		[R/S]	y
7	To change θ or v, go to step 4.			
	To change Δt or g, go to			
	appropriate step, store new			
	value, then go to step 4.			

Example:

Plot the trajectory of a stone cast upwards with a velocity of 20 m/s at an angle of 30° to the horizontal. Use intervals of ¼ second between points plotted. Let g = 9.8 m/s².

Solution:

Keystrokes	Display	
[f] [FIX] 2		
0.25 [STO] 0		
9.8 [STO] 1		
30 [ENTER↑]		
20 [f] [PRGM] [R/S]	0.25	(t_1)
	4.33	(x_1)
[R/S]	2.19	(y_1)
[R/S]	0.50	(t_2)
	8.66	(x_2)
[R/S]	3.78	(y_2)

Keystrokes	Display	
[R/S]	0.75	(t_3)
	12.99	(x_3)
[R/S]	4.74	(y_3)
etc.		

Continue until y becomes negative.

The table of these results is shown below:

t	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
x	4.33	8.66	12.99	17.32	21.65	25.98	30.31	34.64	38.97
y	2.19	3.78	4.74	5.10	4.84	3.98	2.49	0.40	-2.31

The plot of these (x, y) values is made and the stone's trajectory is seen to be a parabola.

Fun and Games

Count-Down Timer

This program provides a count-down 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:

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

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
STO 2	01– 23 2
0	02– 0
R/S	03– 74
STO 1	04– 23 1
g →H	05– 15 6
RCL 2	06– 24 2
×	07– 61
STO 0	08– 23 0
RCL 1	09– 24 1
R/S	10– 74
1	11– 1
STO − 0	12– 23 41 0
RCL 0	13– 24 0
g INT	14– 15 32
g x=0	15– 15 71

KEY ENTRY	DISPLAY
GTO 02	16– 13 02
GTO 11	17– 13 11
g →H	18– 15 6
x⇄y	19– 21
g →H	20– 15 6
x⇄y	21– 21
−	22– 41
RCL 1	23– 24 1
g →H	24– 15 6
÷	25– 71
g 1/x	26– 15 3
RCL 2	27– 24 2
×	28– 61
R/S	29– 74
GTO 01	30– 13 01

REGISTERS			
R_0 Counter	R_1 Time	R_2 Ca	R_3
R_4	R_5	R_6	R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program.			
2	Input timer constant			
	(try 10000)	Ca	[GSB] 01	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] 1 8	Ca
	To proceed depress		[R/S]	
	Then go to step 3.			

Example:

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

Keystrokes	Display
10000 [GSB] 0 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	Display
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	**Display**
9.5803 [ENTER↑]	
9.5701 [GSB] 18	10,967.7421
[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.

Calendar

This program will calculate the day of the week for a given date, or the number of days between two dates, for any dates from March 1, 1700, to February 28, 2100. The program works by assigning the number 1 to March 1, 1700, and a corresponding number to each succeeding day. When calculating the day of the week, a 0 represents Sunday, 1 Monday, 2 Tuesday, etc.

Thus for month m, day d, year y, the number N assigned to that date is

$$N(m, d, y) = \left[365.25\ g(y,m)\right] + \left[30.6\ f(m)\right] + D - 621049$$

where:

$$g(y, m) = \begin{cases} y - 1 & \text{if } m = 1 \text{ or } 2 \\ y & \text{if } m > 2 \end{cases} \quad \text{and } f(m) = \begin{cases} m + 13 & \text{if } m = 1 \text{ or } 2 \\ m + 1 & \text{if } m > 2 \end{cases}$$

$[m]$ represents the integer function, [g] [INT]. E.g., $[6.34] = 6$.

Note:

For days from March 1, 1700, to February 28, 1800, 2 days must be added to the value for N calculated by the program. For days from March 1, 1800, to February 28, 1900, 1 day must be added.

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
2	01– 2
RCL 1	02– 24 1
f x≤y	03– 14 41
GTO 09	04– 13 09
1	05– 1
+	06– 51
RCL 3	07– 24 3
GTO 15	08– 13 15
1	09– 1
3	10– 3
+	11– 51
RCL 3	12– 24 3
1	13– 1
−	14– 41
3	15– 3
6	16– 6
5	17– 5
.	18– 73
2	19– 2
5	20– 5
×	21– 61
g INT	22– 15 32
x⇄y	23– 21

KEY ENTRY	DISPLAY
3	24– 3
0	25– 0
.	26– 73
6	27– 6
×	28– 61
g INT	29– 15 32
+	30– 51
RCL 2	31– 24 2
+	32– 51
6	33– 6
2	34– 2
1	35– 1
0	36– 0
4	37– 4
9	38– 9
−	39– 41
R/S	40– 74
7	41– 7
÷	42– 71
g FRAC	43– 15 33
7	44– 7
×	45– 61
GTO 00	46– 13 00

REGISTERS			
R_0	R_1 Month	R_2 Day	R_3 Year
R_4	R_5	R_6	R_7 Temporary

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in program			
2	Store month	m	[STO] 1	
	day	d	[STO] 2	
	year	y	[STO] 3	
3	Calculate N(m, d, y)		[GSB] 01	N(m, d, y)
4	For day of week, go to step 8			
5	For days between dates, store			
	first N		[STO] 7	
6	Repeat steps 2 and 3 for			
	second date, then		[RCL] 7 [−]	# Days
7	For a new case, go to step 2.			
8	For day of week (0 = Sunday)		[R/S]	Day (0 ,..., 6)
9	For a new case, go to step 2.			

Examples:

1. What day of the week was July 4, 1776?
2. Find the number of days between March 27, 1948, and April 7, 1975.

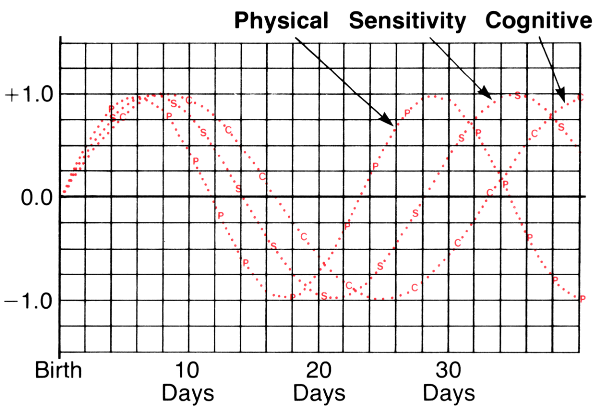
Keystrokes	Display	
[f] [FIX] 0		
7 [STO] 1		
4 [STO] 2		
1776 [STO] 3 [GSB] 01		
2 [+] [R/S]	4.	[Thursday (4).] (Remember to add 2 days.)
3 [STO] 1		
27 [STO] 2		
1948 [STO] 3		
[GSB] 01 [STO] 7	90,607.	
4 [STO] 1		
7 [STO] 2		
1975 [STO] 3		
[GSB] 01	100,479.	
[RCL] 7 [−]	9,872.	

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 ($0 < x \leq 1$) times are regarded as energetic times, you are your most dynamic in the cycle. The low ($-1 \leq x < 0$) 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.
- Set the angular mode to radians before running.

KEY ENTRY	DISPLAY	KEY ENTRY	DISPLAY
f CLEAR PRGM	00	RCL 1	25– 24 1
STO 7	01– 23 7	f x≤y	26– 14 41
RCL 5	02– 24 5	GTO 32	27– 13 32
x⇄y	03– 21	1	28– 1
÷	04– 71	+	29– 51
g FRAC	05– 15 33	RCL 3	30– 24 3
2	06– 2	GTO 38	31– 13 38
×	07– 61	1	32– 1
g π	08– 15 73	3	33– 3
×	09– 61	+	34– 51
f SIN	10– 14 7	RCL 3	35– 24 3
ENTER↑	11– 31	1	36– 1
g ABS	12– 15 34	−	37– 41
g x≠0	13– 15 61	RCL 4	38– 24 4
÷	14– 71	×	39– 61
f LST X	15– 14 73	g INT	40– 15 32
EEX	16– 33	x⇄y	41– 21
7	17– 7	RCL 0	42– 24 0
+	18– 51	×	43– 61
EEX	19– 33	g INT	44– 15 32
7	20– 7	+	45– 51
−	21– 41	RCL 2	46– 24 2
×	22– 61	+	47– 51
R/S	23– 74	g RTN	48– 15 12
2	24– 2		

REGISTERS			
R_0 30.6	R_1 M	R_2 D	R_3 Y
R_4 365.25	R_5 $N_1 - N_2$	R_6 N_2	R_7 23,28,33

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	**Initialize**	30.6	[STO] 0	
		365.25	[STO] 4 [g] [RAD]	
3	Store biodate and calculate			
	day #	M	[STO] 1	
		D	[STO] 2	
		Y	[STO] 3 [GSB] 24	N_1*
4	Store day # for biodate		[STO] 6	
5	Store birthdate and calculate			
	day #	M	[STO] 1	
		D	[STO] 2	
		Y	[STO] 3 [GSB] 24	N_2*
6	Calculate the difference in			
	day #		[RCL] 6 [−] [CHS]	$N_1 - N_2$
7	Store the difference		[STO] 5	
8	To calculate biovalues:			
	Physical:	23	[STO] 7 [GSB] 01	P
	Sensitivity:	28	[STO] 7 [GSB] 01	S
	Cognitive:	33	[STO] 7 [GSB] 01	C
9	To find biovalues for the			
	succeeding days, do	1	[STO] [+] 5	
	then go to step 8.			
10	For a new birthdate, go to			
	step 5, for a new biodate, go			
	to step 3.			
	* See calendar program 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	Display	
[f] [FIX] 2		
30.6 [STO] 0		
365.25 [STO] 4		
[g] [RAD] 6 [STO] 1		
29 [STO] 2		
1976 [STO] 3 [GSB] 24	721,977.00	
[STO] 6		
3 [STO] 1		
27 [STO] 2		
1948 [STO] 3		
[GSB] 24	711,656.00	
[RCL] 6 [−] [CHS]		
[STO] 5	10,321.00	
23 [GSB] 01	−1.00	(P, June 29)
28 [GSB] 01	−0.62	(S)
33 [GSB] 01	−1.00	(C)
1 [STO] [+] 5	1.00	
23 [GSB] 01	−0.98	(P, June 30)
28 [GSB] 01	−0.78	(S)
33 [GSB] 01	−0.97	(C)
1 [STO] [+] 5	1.00	
23 [GSB] 01	−0.89	(P, July 1)
28 [GSB] 01	−0.90	(S)
33 [GSB] 01	−0.91	(C)

Moon Landing Simulator

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, free-falling 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 dust 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/sec from a height of 500 feet. The velocity and height are shown in a combined display as –50.0500, the height 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. If a velocity is ever displayed with no fractional part, for example, –15., it means that you have crashed at a speed of 15 feet/sec. In game terms, this means that you have lost; in real-life, it signifies an even less favorable outcome.

You will start the game with 120 units of fuel. You may burn as much or as little of your available fuel as you wish at each step of your descent; burns of zero are quite common. A burn of 5 units will just cancel gravity and hold your speed constant. Any burn over 5 will act to change your speed in an upward direction. You must take care, however, not to burn more fuel than you have; for if you do, no burn at all will take place, and you will free-fall to your doom! The final velocity shown will be your impact velocity (generally rather high). You may display your remaining fuel at any time by recalling R_2.

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:

$$x = x_0 + v_0 t + \frac{1}{2} at^2 \qquad v = v_0 + at \qquad v^2 = v_0^2 + 2a(x - x_0)$$

where x, v, a, and t are distance, velocity, acceleration, and time.

Note:

1. If you crash before running out of fuel, the crash velocity shown will be the velocity before the burn, rather than the impact velocity.
2. Use only integer values for burns. Any decimal entry will cause an error in the display for V.X.

Programming Remarks:

An interesting feature of this program is the simultaneous display of both speed and altitude (V.X), as for example, –50.0500. This is accomplished by storing the speed V and the altitude X in their normal form (–50.00, 500.00), then dividing X by 10,000 (10^4) before combining them. An additional subtlety involves the question of the sign of V, and whether ($X/10^4$) is to be added to or subtracted from V. For example, if $V = -50$ and $X = 500$, we should subtract: $V - (X/10^4)$, in order to generate a display of –50.0500. But if $V = 10$ and $X = 50$, we should add: $V + (X/10^4)$ in order to display 10.0050. Inspection of the program listing, lines 2 through 12, will reveal how a conditional branch was used to resolve the dilemma.

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
f FIX 4	01– 14 11 4
RCL 0	02– 24 0
EEX	03– 33
4	04– 4
÷	05– 71
RCL 1	06– 24 1
g x<0	07– 15 41
GTO 11	08– 13 11
+	09– 51
GTO 13	10– 13 13
x⇄y	11– 21
−	12– 41
R/S	13– 74
RCL 2	14– 24 2
f x≤y	15– 14 41
GTO 34	16– 13 34
R↓	17– 22
STO − 2	18– 23 41 2
5	19– 5
−	20– 41
STO 3	21– 23 3
2	22– 2
÷	23– 71

KEY ENTRY	DISPLAY
RCL 0	24– 24 0
+	25– 51
RCL 1	26– 24 1
+	27– 51
STO 0	28– 23 0
g x<0	29– 15 41
GTO 44	30– 13 44
RCL 3	31– 24 3
STO + 1	32– 23 51 1
GTO 02	33– 13 02
RCL 1	34– 24 1
g x^2	35– 15 0
RCL 0	36– 24 0
1	37– 1
0	38– 0
×	39– 61
+	40– 51
f $\sqrt{x}$	41– 14 0
CHS	42– 32
STO 1	43– 23 1
RCL 1	44– 24 1
f FIX 0	45– 14 11 0
GTO 00	46– 13 00

REGISTERS			
R_0 x	R_1 v	R_2 Fuel	R_3 Acceleration
R_4	R_5	R_6	R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in program			
2	Initialize	X	500 [STO] 0	500.0000
		V	50 [CHS] [STO] 1	−50.0000
		Fuel	120 [STO] 2	120.0000
3	Display initial V.X		[f] [PRGM] [R/S]	−50.0500
4	Key in burn, calculate new			
	speed and distance	Burn	[R/S]	V.X
5	Perform step 4 till you land			
	or crash			
6	To see remaining fuel at any			
	time		[RCL] 2	Fuel
7	To display speed and distance			
	at any time		[f] [PRGM] [R/S]	V.X
8	To start a new game, go to			
	step 2.			

Example:

Keystrokes	Display
500 [STO] 0	
50 [CHS] [STO] 1	
120 [STO] 2	
[f] [PRGM] [R/S]	*−50.0500*
0 [R/S]	*−55.0448*
5 [R/S]	*−55.0393*

(note constant V when burn = 5)

Keystrokes	Display
30 [R/S]	*−30.0350*
0 [R/S]	*−35.0318*
0 [R/S]	*−40.0280*
0 [R/S]	*−45.0238*
0 [R/S]	*−50.0190*

Keystrokes	Display	
[RCL] 2	85.0000	(remaining fuel)
[f] [PRGM] [R/S]	-50.0190	(display V.X again)
10 [R/S]	-45.0143	
0 [R/S]	-50.0095	
[RCL] 2	75.0000	
10 [R/S]	-45.0048	
25 [R/S]	-25.0013	
20 [R/S]	-25.	Oops.

Nimb

The game of Nimb begins with a collection of N objects, or as the calculator plays it, with the positive number N. Each player alternately subtracts one, two, or more from the total until only one is left. The player forced to take the last one loses.

To begin the game, you must tell the machine how many objects to start with, i.e., the value of N, and specify the maximum number that can be taken in a single move. After each move the machine will display the remaining total. A negative sign indicates that it is the player's move next, while a positive display indicates that it is the HP-33E's move.

As the challenger you are allowed to make the first move. It is possible to win but of course the HP-33E is a master player: it will not let you make an error and win. If you cheat by taking more than the specified limit the calculator will catch you and force you to repeat the move.

This program is based on an HP-25 program by *James L. Horn*.

- Be sure to store one more than the allowable maximum move in Register 1.

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
f FIX 0	01– 14 11 0
STO 0	02– 23 0
CHS	03– 32
R/S	04– 74
+	05– 51
g x<0	06– 15 41
GTO 10	07– 13 10
RCL 3	08– 24 3
GTO 00	09– 13 00
f LST x	10– 14 73
1	11– 1
f x>y	12– 14 51
GTO 32	13– 13 32
R↓	14– 22
RCL 1	15– 24 1
f x≤y	16– 14 41
GTO 32	17– 13 32
x⇄y	18– 21

KEY ENTRY	DISPLAY
STO − 0	19– 23 41 0
RCL 0	20– 24 0
R/S	21– 74
1	22– 1
−	23– 41
RCL 1	24– 24 1
÷	25– 71
g FRAC	26– 15 33
RCL 1	27– 24 1
×	28– 61
g x=0	29– 15 71
1	30– 1
STO − 0	31– 23 41 0
RCL 0	32– 24 0
g x≠0	33– 15 61
GTO 03	34– 13 03
RCL 2	35– 24 2
f FIX 1	36– 14 11 1
GTO 00	37– 13 00

REGISTERS			
R_0 Total	R_1 Max + 1	R_2 3507.1	R_3 55178
R_4	R_5	R_6	R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	Store maximum allowable			
	move plus one	Max. + 1	[STO] 1	
3	Initialize	3507.1	[STO] 2	
		55178	[STO] 3 [f] [PRGM]	
4	Input total number of objects			
	(usually 15)	N.	[R/S]	–N.
5	If number in display is			
	negative, key in your move	Your Move	[R/S]	+Total
6	If number in display is			
	positive, let HP-33E move		[R/S]	–Total
7	Perform steps 4 and 5 until			
	game is over.			
8	At end of game turn			
	calculator upside down to			
	read message.			
9	For another game go to step 4			
10	To change max. move, do			
	step 2.			

Example:

N = 15, Max. move is 3

Keystrokes	Display	
4 [STO] 1		
3507.1 [STO] 2		
55178 [STO] 3		
[f] [PRGM]		
15 [R/S]	–15.	
3 [R/S]	12.	Player takes 3
[R/S]	–9.	HP-33E takes 3

Keystrokes	Display	
5 [R/S]	-9.	Player tries to cheat
2 [R/S]	7.	Player takes 2
[R/S]	-5.	HP-33E takes 2
3 [R/S]	2.	Player takes 3
[R/S]	-1.	HP-33E takes 1
1 [R/S]	55,178.	Player takes last one

Turn calculator upside down for message (BLISS).

Finance

Because several of the finance programs have certain quantities in common, a word about these variables and the names used to refer to them may be helpful.

Five main variables recur in finance problems: n, i, PMT, PV, and FV. The first of these, n, denotes the total number of periods. The periodic interest rate i must be expressed in these programs as a decimal. Thus an annual interest rate of 6% is expressed as 0.06, which as a monthly rate would be 0.06/12 = 0.005. PMT refers to the amount of the periodic payment. The present value, PV, is the value occurring at the beginning of the first period, while the future value, FV, is the value at the end of the last period.

These programs employ the convenient sign convention used in the most recent HP calculators and programs. Cash received is represented by a positive value (+). Cash paid out is represented by a negative value (−).

Mortgage Loan Accumulated Interest/Remaining Balance

This program will allow the user to calculate the amount paid to interest, for one payment or over a number of payments, as well as the amount of principal still unpaid, i.e., the remaining balance. The user must input the following values: the initial amount of the loan, the periodic interest rate, and the periodic payment amount. He must then key in a beginning payment number, J, and an ending payment number, K. The program will compute the accumulated interest charge from payment J through payment K, inclusive, and the balance remaining after payment K. If one wishes to find the amount of interest paid in a single payment, he can simply set K = J.

The program can also be used to generate a limited amortization schedule showing the balance remaining after successive payments. This can be done by leaving J = 1 and increasing K by 1 at each iteration. Outputs will be the total amount paid to interest over the first K payments, and the balance remaining after payment K.

Equations:

$$BAL_K = \frac{1}{(1+i)^{-K}} \left[PMT \frac{(1+i)^{-K} - 1}{i} - PV \right]$$

$$Int_{J-K} = BAL_K - BAL_{J-1} + (K - J + 1)\ PMT$$

where: BAL_n = remaining balance after payment n

Int_{J-K} = accumulated interest, payments J through K

PV = initial loan amount

PMT = periodic payment amount

i = periodic interest rate

Notes:

1. The periodic interest rate i must be entered as a decimal. For example, for monthly payments with an annual interest rate of 9%, the periodic interest rate should be input as $i = \frac{.09}{12} = 0.0075$.
2. The use of this program is not restricted to mortgage loans, but applies equally well to any loan which is being repaid with equal periodic payments.
3. Cash received is represented by a positive value (+). Cash paid out is represented by a negative value (−).

KEY ENTRY	DISPLAY	KEY ENTRY	DISPLAY
f CLEAR PRGM	00	×	19– 61
RCL 1	01– 24 1	+	20– 51
1	02– 1	R/S	21– 74
+	03– 51	x⇄y	22– 21
STO 0	04– 23 0	GTO 00	23– 13 00
RCL 5	05– 24 5	CHS	24– 32
GSB 24	06– 12 24	f y^x	25– 14 3
RCL 0	07– 24 0	STO 7	26– 23 7
RCL 4	08– 24 4	1	27– 1
1	09– 1	−	28– 41
−	10– 41	RCL 1	29– 24 1
GSB 24	11– 12 24	÷	30– 71
−	12– 41	RCL 2	31– 24 2
RCL 5	13– 24 5	×	32– 61
RCL 4	14– 24 4	RCL 3	33– 24 3
−	15– 41	−	34– 41
1	16– 1	RCL 7	35– 24 7
+	17– 51	÷	36– 71
RCL 2	18– 24 2	g RTN	37– 15 12

REGISTERS			
R_0 1 + i	R_1 i	R_2 PMT	R_3 PV
R_4 J	R_5 K	R_6	R_7 $(1 + i)^{-n}$

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in program			
2	Store the following variables:			
	Periodic interest (decimal)	i	STO 1	
	Periodic payment	PMT*	STO 2	
	Initial loan amount	PV*	STO 3	
	Starting payment number	J	STO 4	
	Ending payment number	K	STO 5	
3	Compute accumulated interest			
	from payments J through K.		GSB 01	Int_{J-K}
4	Display remaining balance			
	after payment K.		R/S	BAL_K
5	To change any variable, store			
	the new value in the			
	appropriate register and go to			
	step 3.			
	* Note: Cash received is			
	represented by a positive			
	value (+). Cash paid out is			
	represented by a negative			
	value (−).			

Example:

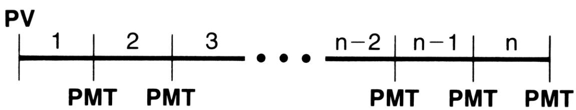
A mortgage is arranged so that the first payment is made at the end of October, 1974 (i.e., October is payment period 1). It is a $25,000 loan at 8% with monthly payments of $200. What is the accumulated interest for 1974 (periods 1—3) and for 1975 (periods 4—15) and what balance remains at the end of each year?

Solution:

(Notice that i must be entered as a decimal, monthly rate.)

Keystrokes	Display	
f FIX 2		
.08 ENTER↑ 12 ÷		
STO 1		
200 CHS		
STO 2		
25000 STO 3		
1 STO 4		
3 STO 5 GSB 01	-499.33	(interest paid in 1974)
R/S	-24,899.33	(remaining balance at end of 1974)
4 STO 4		
15 STO 5 R/S	-1,976.65	(interest paid in 1975)
R/S	-24,475.98	(remaining balance at end of 1975)

Mortgage Loan
Payment, Present Value, Number Of Periods



For a loan which is being repaid with equal periodic payments, this program will calculate the payment amount, the present value, or the number of periods of the loan, given the periodic interest rate and the two other variables.

Remember that the periodic interest rate i must be expressed as a decimal, e.g., 6% is represented as 0.06.

The equations used are as follows:

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

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

$$n = -\frac{\ln\left(1 - i\left|PV/PMT\right|\right)}{\ln(1+i)}$$

Note:

Cash received is represented by a positive value (+). Cash paid out is represented by a negative value (−).

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
RCL 2	01– 24 2
GSB 31	02– 12 31
÷	03– 71
RCL 4	04– 24 4
CHS	05– 32
×	06– 61
GTO 00	07– 13 00
GSB 31	08– 12 31
RCL 2	09– 24 2
÷	10– 71
RCL 3	11– 24 3
CHS	12– 32
×	13– 61
GTO 00	14– 13 00
1	15– 1
RCL 4	16– 24 4
RCL 3	17– 24 3
÷	18– 71
g ABS	19– 15 34
RCL 2	20– 24 2
×	21– 61
−	22– 41
f LN	23– 14 1
RCL 2	24– 24 2
1	25– 1
+	26– 51
f LN	27– 14 1
÷	28– 71
CHS	29– 32
GTO 00	30– 13 00
1	31– 1
RCL 2	32– 24 2
1	33– 1
+	34– 51
RCL 1	35– 24 1
CHS	36– 32
f y^x	37– 14 3
−	38– 41
g RTN	39– 15 12

REGISTERS			
R_0	R_1 n	R_2 i	R_3 PMT
R_4 PV	R_5	R_6	R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	For payment: no. of periods	n	[STO] 1	
	Interest Rate	i	[STO] 2	
	Present Value	PV*	[STO] 4	
			[GSB] 01	PMT*
3	For present value:			
	No. of Periods	n	[STO] 1	
	Interest Rate	i	[STO] 2	
	Payment	PMT*	[STO] 3	
			[GSB] 08	PV*
4	For number of payments:			
	Interest Rate	i	[STO] 2	
	Payment	PMT*	[STO] 3	
	Present Value	PV*	[STO] 4	
			[GSB] 15	n
5	For a new case, go to step			
	2, 3 or 4.			
	* Note: Cash received is			
	represented by a positive			
	value (+). Cash paid out is			
	represented by a negative			
	value (−).			

Example 1:

What monthly payment is required to amortize a $3000 loan at 9.5% (0.095) in 36 months?

Keystrokes	Display	
f FIX 2		
36 STO 1		
.095 ENTER↑		
12 ÷ STO 2		
3000 STO 4 GSB 01	-96.10	($)

Example 2:

You are willing to pay $175 per month for 24 months on a 9.5% loan. How much can you borrow?

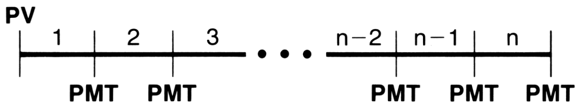
Keystrokes	Display	
24 STO 1		
.095 ENTER↑		
12 ÷ STO 2		
175 CHS STO 3		
GSB 08	3,811.43	($)

Example 3:

How many months will it take to pay off a $4000 loan if your monthly payment is $200 and the annual interest rate is 9.5%.

Keystrokes	Display	
.095 ENTER↑		
12 ÷ STO 2		
200 CHS STO 3		
4000 STO 4		
GSB 15	21.86	(Months)

Mortgage Loan Interest Rate



This program will calculate the interest rate on a loan with equal periodic payments. The user must specify the number of periods, the present value or initial loan amount, and the payment amount.

The program performs an iterative solution for i using Newton's method:

$$i_{k+1} = i_k - \frac{f(i_k)}{f'(i_k)}$$

where:

$$f(i) = \frac{1 - (1 + i)^{-n}}{i} - \left| \frac{PV}{PMT} \right|$$

The initial guess for i is given by

$$i_0 = \left| \frac{PMT}{PV} \right| - \frac{1}{n^2} \left| \frac{PV}{PMT} \right|$$

Note:

Cash received is represented by a positive value (+). Cash paid out is represented by a negative value (−).

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
RCL 3	01– 24 3
g ABS	02– 15 34
ENTER↑	03– 31
g 1/x	04– 15 3
x⇄y	05– 21
RCL 1	06– 24 1
g x^2	07– 15 0
÷	08– 71
−	09– 41
STO 2	10– 23 2
RCL 3	11– 24 3
g ABS	12– 15 34
RCL 2	13– 24 2
×	14– 61
1	15– 1
RCL 2	16– 24 2
1	17– 1
+	18– 51
RCL 1	19– 24 1
CHS	20– 32
f y^x	21– 14 3
STO 5	22– 23 5
−	23– 41
−	24– 41

KEY ENTRY	DISPLAY
RCL 1	25– 24 1
RCL 2	26– 24 2
g 1/x	27– 15 3
1	28– 1
+	29– 51
÷	30– 71
1	31– 1
+	32– 51
RCL 5	33– 24 5
×	34– 61
1	35– 1
−	36– 41
RCL 2	37– 24 2
÷	38– 71
÷	39– 71
STO + 2	40– 23 51 2
g ABS	41– 15 34
EEX	42– 33
6	43– 6
CHS	44– 32
f x≤y	45– 14 41
GTO 11	46– 13 11
RCL 2	47– 24 2
GTO 00	48– 13 00

REGISTERS			
R_0	R_1 n	R_2 i	R_3 PV/PMT
R_4 $(1+i)^{-n}$	R_5 used	R_6	R_7

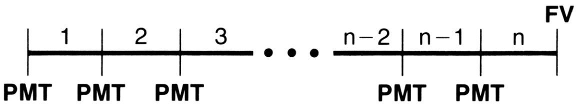
STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in the program			
2	Store number of payments	n	[STO] 1	
3	Key in present value and			
	payment amount	PV*	[ENTER↑]	
		PMT*	[÷] [STO] 3	PV/PMT
4	Calculate interest		[GSB] 01	i (decimal)
				Monthly Rate
		12	[EEX] 2 [×]	i (%) Annual
				Rate
5	For a new case, go to step 2.			
	* Note: Cash received is			
	represented by a positive			
	value (+). Cash paid out is			
	represented by a negative			
	value (−).			

Example:

You recently obtained a $2500 car loan for 36 months. If your monthly payment is $86.67, what is the annual percentage rate?

Keystrokes	Display	
36 [STO] 1		
2500 [ENTER↑] 86.67		
[CHS] [÷] [STO] 3		
[GSB] 01	0.0125	Monthly Rate
[f] [FIX] 2		
12 [EEX] 2 [×]	15.01	% Annual Rate

Periodic Savings
Payment, Future Value, Number Of Periods



This program calculates payment, future value, or number of time periods for a schedule of periodic payments into a savings account, given the interest rate and two of the three other variables. Remember that i must be input as a decimal, e.g., 6% is expressed as 0.06.

Then n, PMT, or FV may be calculated from the following formulas:

$$n = \frac{\ln \left[\left| \frac{FV}{PMT} \right| i + (1 + i) \right]}{\ln (1 + i)} - 1$$

$$PMT = \frac{-FV\,i}{(1 + i)^{n+1} - (1 + i)}$$

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

Note:

Cash received is represented by a positive value (+). Cash paid out is represented by a negative value (−).

KEY ENTRY	DISPLAY
f CLEAR PRGM	00
RCL 2	01– 24 2
RCL 5	02– 24 5
×	03– 61
RCL 3	04– 24 3
÷	05– 71
g ABS	06– 15 34
RCL 2	07– 24 2
1	08– 1
+	09– 51
STO 0	10– 23 0
+	11– 51
f LN	12– 14 1
RCL 0	13– 24 0
f LN	14– 14 1
÷	15– 71
1	16– 1
−	17– 41
GTO 00	18– 13 00
RCL 5	19– 24 5
CHS	20– 32
RCL 2	21– 24 2
×	22– 61
RCL 2	23– 24 2

KEY ENTRY	DISPLAY
1	24– 1
+	25– 51
÷	26– 71
GSB 41	27– 12 41
÷	28– 71
GTO 00	29– 13 00
RCL 3	30– 24 3
CHS	31– 32
RCL 2	32– 24 2
1	33– 1
+	34– 51
×	35– 61
GSB 41	36– 12 41
×	37– 61
RCL 2	38– 24 2
÷	39– 71
GTO 00	40– 13 00
f LST X	41– 14 73
RCL 1	42– 24 1
f y^x	43– 14 3
1	44– 1
−	45– 41
g RTN	46– 15 12

REGISTERS			
R_0 (1 + i)	R_1 n	R_2 i	R_3 PMT
R_4	R_5 FV	R_6	R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in program.			
2	**To calculate number of**			
	payments	i (decimal)	STO 2	
		PMT*	STO 3	
		FV*	STO 5	
			GSB 01	n
3	**To calculate periodic payment**			
	amount	n	STO 1	
		i (decimal)	STO 2	
		FV*	STO 5	
			GSB 19	PMT*
4	**To calculate future value**	n	STO 1	
		i (decimal)	STO 2	
		PMT*	STO 3	
			GSB 30	FV*
5	For a new case, go to step 2,			
	3, or 4.			
	* Note: Cash received is			
	represented by a positive			
	value (+). Cash paid out is			
	represented by a negative			
	value (−).			

Example 1:

How long will it take to save $15,000 if you are making quarterly deposits of $400 at 6% annual interest?

Keystrokes	Display	
[f] [FIX] 2		
.06 [ENTER↑]		
4 [÷] [STO] 2		
400 [CHS] [STO] 3		
15000 [STO] 5		
[GSB] 01	**29.62**	Quarters (7.40 years)

Example 2:

You will need $10,000 in 7 years. How large a monthly payment do you need to make if the annual interest rate is 6½%?

Keystrokes	Display	
7 [ENTER↑]		
12 [×] [STO] 1		
.065 [ENTER↑]		
12 [÷] [STO] 2		
10000 [STO] 5		
[GSB] 19	**–93.82**	($)

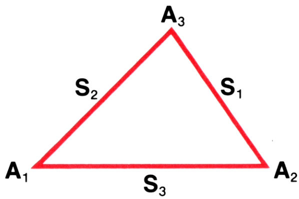
Example 3:

How much money will a person have if he deposits $150 at the end of each month for a period of 3 years? He receives 6% annual interest.

Keystrokes	Display	
3 [ENTER↑]		
12 [×] [STO] 1		
.06 [ENTER↑]		
12 [÷] [STO] 2		
150 [CHS] [STO] 3		
[GSB] 30	**5,929.92**	($)

Triangle Solutions

These programs may be used to find the sides, the angles, and the area of plane triangles.



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 these programs 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 since two solutions exist if the angle is less than 90° and the adjacent side is larger than the side opposite), two angles and the adjacent side (AAS) and three sides (SSS).

Two programs are used to solve for these cases:

Program A:	SSA, AAS, ASA & SAS
Program B:	SSS & SAS

The results are stored in storage registers 1 through 6 as follows:

Side 1	Register 1
Side 2	Register 2
Side 3	Register 3
Angle 1	Register 4
Angle 2	Register 5
Angle 3	Register 6

These programs are based on HP-25 programs by William R. Hewlett.

Remarks:

- Any angular mode may be used.
- Angles must be entered as decimals. The [g] [→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.
- The program does not automatically determine whether 2 solutions exist for the SSA case. The user can determine by inspection; if the angle is less than 90° and the adjacent side is longer than the side opposite, 2 solutions exist. Attempts to find a nonexistent second solution result in negative values for some of the parameters.

Program A: SSA, AAS, ASA & SAS

KEY ENTRY	DISPLAY	KEY ENTRY	DISPLAY
f CLEAR PRGM	00	×	25– 61
RCL 4	01– 24 4	STO 2	26– 23 2
f SIN	02– 14 7	RCL 4	27– 24 4
RCL 3	03– 24 3	RCL 2	28– 24 2
×	04– 61	f →R	29– 14 4
RCL 1	05– 24 1	RCL 3	30– 24 3
÷	06– 71	x⇄y	31– 21
g SIN⁻¹	07– 15 7	−	32– 41
R/S	08– 74	g →P	33– 15 4
STO 6	09– 23 6	STO 1	34– 23 1
RCL 6	10– 24 6	x⇄y	35– 21
RCL 4	11– 24 4	STO 5	36– 23 5
+	12– 51	RCL 4	37– 24 4
STO 5	13– 23 5	+	38– 51
GTO 19	14– 13 19	f COS	39– 14 8
RCL 5	15– 24 5	CHS	40– 32
RCL 4	16– 24 4	g COS⁻¹	41– 15 8
+	17– 51	STO 6	42– 23 6
STO 6	18– 23 6	f SIN	43– 14 7
RCL 5	19– 24 5	×	44– 61
f SIN	20– 14 7	RCL 2	45– 24 2
RCL 6	21– 24 6	×	46– 61
f SIN	22– 14 7	2	47– 2
÷	23– 71	÷	48– 71
RCL 3	24– 24 3	GTO 00	49– 13 00

REGISTERS			
R_0	R_1 $SIDE_1$	R_2 $SIDE_2$	R_3 $SIDE_3$
R_4 ANG_1	R_5 ANG_2	R_6 ANG_3	R_7

Program B: SSS & SAS

KEY ENTRY	DISPLAY	KEY ENTRY	DISPLAY
f CLEAR PRGM	00	RCL 3	20– 24 3
RCL 2	01– 24 2	x⇄y	21– 21
g x^2	02– 15 0	−	22– 41
RCL 3	03– 24 3	g →P	23– 15 4
g x^2	04– 15 0	STO 1	24– 23 1
+	05– 51	x⇄y	25– 21
RCL 1	06– 24 1	STO 5	26– 23 5
g x^2	07– 15 0	RCL 4	27– 24 4
−	08– 41	+	28– 51
RCL 2	09– 24 2	f COS	29– 14 8
RCL 3	10– 24 3	CHS	30– 32
×	11– 61	g COS^{-1}	31– 15 8
2	12– 2	STO 6	32– 23 6
×	13– 61	f SIN	33– 14 7
÷	14– 71	×	34– 61
g COS^{-1}	15– 15 8	RCL 2	35– 24 2
STO 4	16– 23 4	×	36– 61
RCL 4	17– 24 4	2	37– 2
RCL 2	18– 24 2	÷	38– 71
f →R	19– 14 4	GTO 00	39– 13 00

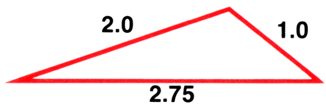
REGISTERS			
R_0	R_1 $SIDE_1$	R_2 $SIDE_2$	R_3 $SIDE_3$
R_4 ANG_1	R_5 ANG_2	R_6 ANG_3	R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Key in program A or B.			
	Program A			
2	SSA (2 sides and adjacent			
	angle)			
	Side 1	S_1	[STO] 1	
	Side 3	S_3	[STO] 3	
	Angle 1	A_1	[STO] 4	
	a) to start calculation		[GSB] 01	
	b) at 1st halt		[R/S]	Area
	c) review registers			Solution I*
	d) for second solution if it			
	exists		[GSB] 01	
	e) at 1st halt		[CHS] [R/S]	Area
	f) review registers			Solution II*
3	AAS (2 angles and adjacent			
	side)			
	Angle 1	A_1	[STO] 4	
	Angle 3	A_3	[STO] 6	
	Side 3	S_3	[STO] 3	
			[GSB] 10	Area
	review registers			Solution*
4	ASA (2 angles and included			
	side)			
	Angle 1	A_1	[STO] 4	
	Side 3	S_3	[STO] 3	
	Angle 2	A_2	[STO] 5	
			[GSB] 15	Area
	review registers			Solution*

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
5	SAS (2 sides and included			
	angle)			
	Side 2	S_2	[STO] 2	
	Angle 1	A_1	[STO] 4	
	Side 3	S_3	[STO] 3	
			[GSB] 27	Area
	review registers			Solution*
	Program B			
6	SSS (3 sides)			
	Side 1	S_1	[STO] 1	
	Side 2	S_2	[STO] 2	
	Side 3	S_3	[STO] 3	
			[GSB] 01	Area
	review registers			Solution*
7	SAS (2 sides and included			
	angle)			
	Side 2	S_2	[STO] 2	
	Angle 1	A_1	[STO] 4	
	Side 3	S_3	[STO] 3	
			[GSB] 17	Area
	review registers			Solution*
	* Area of the triangle is			
	displayed. Sides and angles			
	are obtained by reviewing			
	registers.			

Example 1:

Find the angles and the area for the following triangle.



SSS: Use program "B"

Keystrokes	Display	
1 [STO] 1		
2 [STO] 2		
2.75 [STO] 3		
[GSB] 01	*0.7679*	(Area)
[RCL] 1	*1.0000*	(S_1)
[RCL] 2	*2.0000*	(S_2)
[RCL] 3	*2.7500*	(S_3)
[RCL] 4	*16.2136*	(A_1)
[RCL] 5	*33.9479*	(A_2)
[RCL] 6	*129.8384*	(A_3)

Example 2:

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_3 = 19.6 ft., A_1 = 61.06° and A_3 = 40.25°.

AAS: Use program "A"

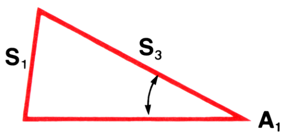
Keystrokes	Display	
61.06 [STO] 4		
19.6 [STO] 3		
40.25 [STO] 6		
[GSB] 10	*255.1059*	(Area, ft^2)

Keystrokes	Display	
RCL 1	26.5467	(S_1, ft)
RCL 2	29.7456	(S_2, ft)
RCL 3	19.6000	(S_3, ft)
RCL 4	61.0600	(A_1, deg)
RCL 5	78.6900	(A_2, deg)
RCL 6	40.2500	(A_3, deg)

Example 3:

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

$Side_1 = 25.6$
$Side_3 = 32.8$
$Angle_1 = 42.3°$



Note:

Since $S_3 > S_1$ and $A_1 < 90°$, 2 solutions exist.

SSA: Use Program "A"

Keystrokes	Display	
25.6 STO 1		
32.8 STO 3		
42.3 STO 4		
GSB 01		
R/S (Solution #1)	410.8542	(Area)
RCL 1	25.6000	(S_1)
RCL 2	37.2238	(S_2)
RCL 3	32.8000	(S_3)
RCL 4	42.3000	(A_1)
RCL 5	78.1245	(A_2)
RCL 6	59.5755	(A_3)

Keystrokes	Display	
GSB 01 (Solution #2)		
CHS R/S	124.6785	(Area)
RCL 1	25.6000	(S_1)
RCL 2	11.2960	(S_2)
RCL 3	32.8000	(S_3)
RCL 4	42.3000	(A_1)
RCL 5	17.2755	(A_2)
RCL 6	120.4245	(A_3)

HEWLETT 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 800/992-5710.)

00033-90024 Printed in U.S.A.