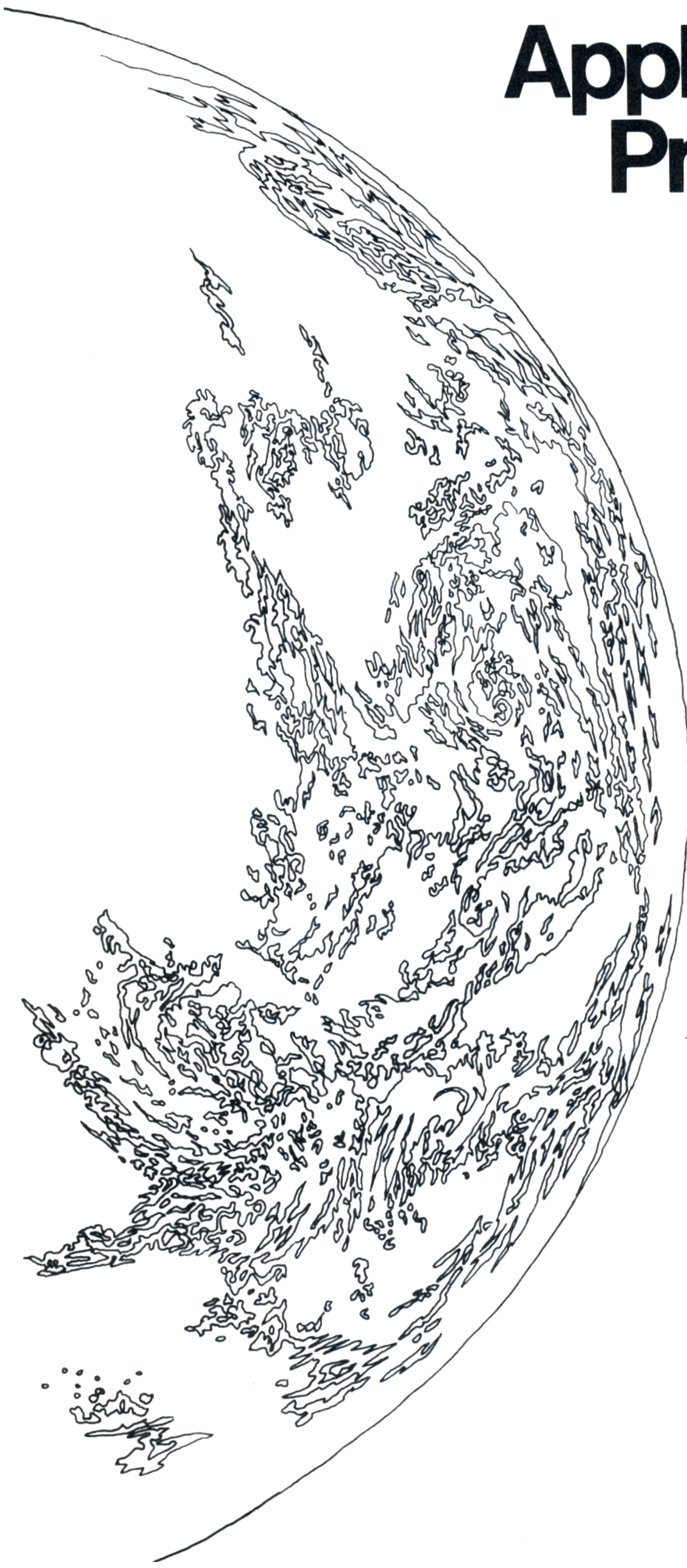
HEWLETT-PACKARD

HP-25

Applications Programs



00025-90011 Rev. B 7/75

INTRODUCTION

Welcome. You are about to step into a field that, ten years ago, was open only to users of large computer systems costing tens or hundreds of thousands of dollars, and even five years ago, required a several-thousand-dollar calculator that occupied the better part of a desktop. Today, the HP-25 puts programming into the hands of the individual. It is hoped that this book will allow you to realize some of the potential of this calculating instrument.

These HP-25 Applications Programs have been drawn from the varied fields of mathematics, statistics, finance, surveying, navigation, and games. They have been arranged in eight chapters which follow roughly the above classification. Each program is furnished with a full explanation which includes a description of the problem, any pertinent equations, a list of keystrokes to be entered into program memory, a set of instructions for running the program, and an example or two, with solutions. To use the programs does not require any proficiency in programming, but some familiarity with the HP-25 Owner's Handbook is assumed.

For users who want to enhance their understanding of programming principles and techniques, a number of programs are provided to help in this respect. The first program in each chapter contains, in addition to the usual explanations, a more detailed description of the problem, a commented list of the program keystrokes with a step-by-step tracing of the contents of the stack registers, and a list of the keystrokes required to solve the example problem. Whenever an interesting programming technique is used in one of these programs, it is described in a short section headed "Programming Remarks", which, if present, will immediately precede the list of program keystrokes.

Thus, whether your interest lies in solving a particular problem in a specific area, or in learning more about the programming power of your calculator, we hope that this book will help you get the most from your HP-25.

TABLE OF CONTENTS

A WORD ABOUT PROGRAM USAGE

Various kinds of information are provided to explain the use of each program. Besides a short description of the problem, a list of applicable equations, and an example problem with solution, there are two forms that deserve some explanation: the Program form and the User Instructions form.

Two different Program forms are provided, one of which is just a simplified version of the other. The detailed form is used for a total of eight programs, one per chapter, with the simpler form serving for the rest. A section of a detailed form, taken from the Plotting/Graphing program in Chapter 1, is shown below:

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	COMMENTS	REGISTERS
00			v	θ				R_0 Δt
01	14 09	f →R	v_x	v_y			Use polar-to-rectangular for	
02	23 02	STO 2	v_x	v_y			$v_x = v \cos\theta$ = horiz. vel.	
03	21	x⇄y	v_y	v_x				R_1 g
04	23 03	STO 3	v_y	v_x			$v_y = v \sin\theta$ = vert. vel.	
05	00	0	0					
06	23 04	STO 4	0				Initialize: $t = 0$	R_2 v_x
07	24 00	RCL 0	Δt				Start of loop	
08	23 51 04	STO + 4	Δt				Next time interval:	
09	24 04	RCL 4	t				$t \leftarrow t + \Delta t$	R_3 v_y
10	15 02	g x^2	t^2					

The rightmost column, headed REGISTERS, explains what variables are stored in storage registers R_0 through R_7. The rest of the form is divided into eight columns. The first two columns describe the appearance of the display as the program is being keyed in: LINE shows the step number for the current instruction and CODE denotes the numeric keycodes corresponding to the keystrokes in the next column, KEY ENTRY. The entries in this column are the keys that must be pressed to enter the program into program memory. The **ENTER↑** key is denoted in this column as [↑]; all other key designations are identical to those appearing on the HP-25.

The next four columns, X, Y, Z, and T, trace the contents of the stack registers as they would change during execution of the program in RUN mode. Each entry under X, Y, Z, or T gives the contents of the respective register *after* the instruction on that line has been executed. The COMMENTS column contains additional step-by-step explanation of the program's calculations.

These last columns, X, Y, Z, T, and COMMENTS, are provided to help the interested user acquire a detailed, in-depth understanding of a particular program, or of programming techniques in general.

The simplified Program forms contain the same information as the detailed forms except for the omission of columns X, Y, Z, T, and COMMENTS.

The User Instructions form is the user's guide to operating the program to solve his own particular problem. This form, which is composed of five columns, is illustrated below for the same program from Chapter 1, 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	↑				
		v	f	PRGM			
5	Perform steps 5 and 6 any num-						
	ber of times: Display time and		R/S				(t)
	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.						

Reading from left to right, the STEP column gives the instruction step number. The INSTRUCTIONS column gives instructions and comments concerning the operations to be performed. Steps are executed in sequential order except where the INSTRUCTIONS column directs otherwise.

Normally, the first instruction is "Key in program", which means to store the keystrokes of the program in program memory (switch to PRGM mode, press [f] [PRGM], key in the program, then switch back to RUN mode).

Repeated processes, used in most cases for a long string of input/output data, are outlined with a bold border, as in steps 5 and 6 above. In this case, the steps are repeated in order to generate a number of (x,y) pairs for a graph.

The INPUT DATA/UNITS column specifies the input data to be supplied, and the units of data if applicable. The KEYS column specifies the keys to be pressed. [↑] is used for the [ENTER↑] key, and all other key designations are identical to those appearing on the HP-25. Ignore any blank positions in the KEYS column.

Some programs are complex enough that users have to press additional keys to generate some results. Those keys are also shown in the KEYS column.

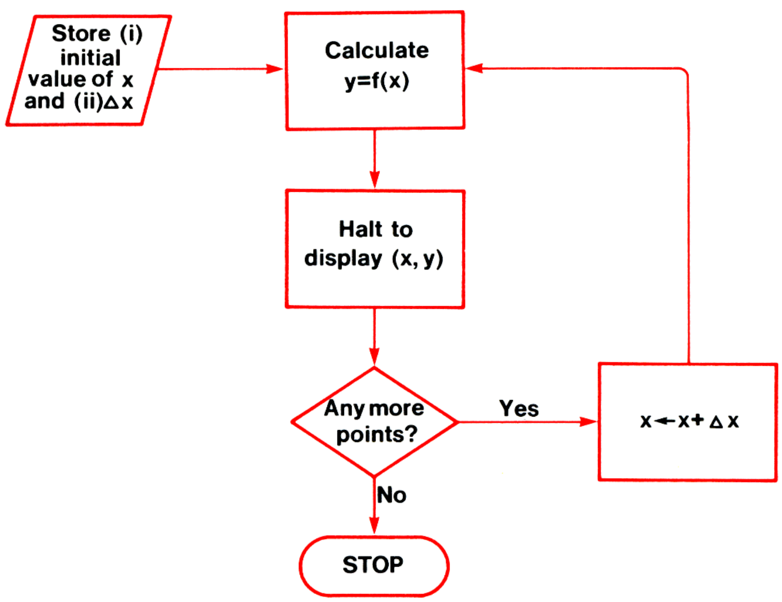
The OUTPUT DATA/UNITS column shows intermediate and final results that have been calculated either from the keyboard or from an executing program, and the units of data if applicable. Parentheses around an output variable, such as (t) in step 5, indicate that the result is displayed only briefly by a PAUSE instruction ([f] [PAUSE]).

CHAPTER 1 ALGEBRA AND NUMBER THEORY

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-25 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.

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS
LINE	CODE						
00			v	θ			
01	14 09	f →R	v_x	v_y			Use polar-to-rectangular for
02	23 02	STO 2	v_x	v_y			$v_x = v \cos \theta$ = horiz. vel.
03	21	x⇄y	v_y	v_x			
04	23 03	STO 3	v_y	v_x			$v_y = v \sin \theta$ = vert. vel.
05	00	0	0				
06	23 04	STO 4	0				Initialize: t = 0
07	24 00	RCL 0	Δt				Start of loop
08	23 51 04	STO + 4	Δt				Next time interval:
09	24 04	RCL 4	t				$t \leftarrow t + \Delta t$
10	15 02	g x^2	t^2				
11	24 01	RCL 1	g	t^2			
12	61	x	$g\,t^2$				
13	02	2	2	$g\,t^2$			
14	71	÷	$1/2\,g\,t^2$				
15	32	CHS	$-1/2\,g\,t^2$				
16	24 04	RCL 4	t	$-1/2\,g\,t^2$			
17	24 03	RCL 3	v_y	t	$-1/2\,g\,t^2$		
18	61	x	$v_y\,t$	$-1/2\,g\,t^2$			
19	51	+	y				$y = v_y\,t - 1/2\,g\,t^2$
20	24 04	RCL 4	t	y			
21	24 02	RCL 2	v_x	t	y		
22	61	x	x	y			$x = v_x\,t$
23	24 04	RCL 4	t	x	y		
24	14 74	f PAUSE	t	x	y		Pause to display t
25	22	R↓	x	y		t	
26	74	R/S	x	y		t	Halt and display x
27	21	x⇄y	y	x		t	
28	74	R/S	y	x		t	Halt and display y
29	13 07	GTO 07	y	x		t	Branch back for next t
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							

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	↑				
		v	f	PRGM			
5	Perform steps 5 and 6 any num-						
	ber of times: Display time and		R/S				(t)
	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/s^2.

Solution:

0.25 [STO] [0] 9.8 [STO] [1] 30 [↑] 20 [f] [PRGM] [R/S] ⟶ 0.25 (t_1)

4.33 (x_i)

[R/S] ⟶ 2.19 (y_1)

[R/S] ⟶ 0.5 (t_2)

8.66 (x_2)

[R/S] ⟶ 3.78 (y_2)

[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.

QUADRATIC EQUATION

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 computing the root with the larger absolute value:

If

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

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

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

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	31	↑
02	22	R↓
03	71	÷
04	02	2
05	71	÷
06	32	CHS
07	31	↑
08	15 02	g x^2
09	22	R↓
10	22	R↓
11	21	x⇄y
12	71	÷
13	23 00	STO 0
14	41	–
15	14 74	f PAUSE
16	15 41	g x<0
17	13 31	GTO 31
18	14 02	f $\sqrt{x}$
19	21	x⇄y
20	15 41	g x<0
21	13 24	GTO 24
22	51	+
23	13 26	GTO 26
24	21	x⇄y

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	41	–
26	74	R/S
27	15 22	g 1/x
28	24 00	RCL 0
29	61	x
30	13 00	GTO 00
31	32	CHS
32	14 02	f $\sqrt{x}$
33	21	x⇄y
34	74	R/S
35	21	x⇄y
36	13 00	GTO 00
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 c/a
R_1
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM			
3	Enter coefficients and display D	c	↑				
		b	↑				
		a	R/S				(D)
4	If D ⩾ 0, roots are real						x_1
			R/S				x_2
	or						
	If D < 0, roots are complex of						
	form u ± iv						u
			R/S				v
5	For new case, go to step 3.						

Example:

Find solutions to the three equations below:

1. $x^2 + x - 6 = 0$

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

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

Solutions:

1. $D = 6.25$

   $x_1 = -3.00$

   $x_2 = 2.00$

2. $D = 0.44$

   $x_1 = -1.00$

   $x_2 = 0.33$

3. $D = -1.94$

   $x_1, x_2 = 0.75 \pm 1.39\ i$

COMPLEX ARITHMETIC, +, –, x, ÷

Let $a_1 + ib_1$ and $a_2 + ib_2$ be two complex numbers. The arithmetic operations +, –, x, ÷ are defined as follows:

1. +, addition

$$(a_1 + ib_1) + (a_2 + ib_2) = (a_1 + a_2) + (b_1 + b_2)i$$

2. –, subtraction

$$(a_1 + ib_1) - (a_2 + ib_2) = (a_1 - a_2) + (b_1 - b_2)i$$

3. x, multiplication

$$(a_1 + ib_1) \times (a_2 + ib_2) = r_1\ r_2\ e^{i(\theta_1 + \theta_2)}$$

4. ÷, division

$$\frac{(a_1 + ib_1)}{(a_2 + ib_2)} = \frac{r_1}{r_2}\ e^{i(\theta_1 - \theta_2)},\ a_2 + ib_2 \neq 0$$

where $r_1\ e^{i\theta_1}$ is the polar representation of $a_1 + ib_1$ and $r_2\ e^{i\theta_2}$ is the polar representation of $a_2 + ib_2$. In each case let the answer be x + iy.

After a calculation is finished x is stored in R_o as well as the X-register and y is stored in R_1 as well as the Y-register. In this way arithmetic operations can be chained together.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	32	CHS
02	21	x⇄y
03	32	CHS
04	21	x⇄y
05	24 00	RCL 0
06	51	+
07	21	x⇄y
08	24 01	RCL 1
09	51	+
10	13 31	GTO 31
11	15 09	g →P
12	15 22	g 1/x
13	21	x⇄y
14	32	CHS
15	21	x⇄y
16	13 18	GTO 18
17	15 09	g →P
18	23 02	STO 2
19	22	R↓
20	24 01	RCL 1
21	24 00	RCL 0
22	15 09	g →P
23	24 02	RCL 2
24	61	x

DISPLAY		KEY ENTRY
LINE	CODE	
25	23 02	STO 2
26	22	R↓
27	51	+
28	24 02	RCL 2
29	14 09	f →R
30	21	x⇄y
31	23 01	STO 1
32	21	x⇄y
33	23 00	STO 0
34	13 00	GTO 00
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 a_1, x
R_1 b_1, y
R_2 Used
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store first complex number	b_1	STO	1			
		a_1	STO	0			
3	Key in next number	b_2	↑				
		a_2					
4	For addition		GTO	05	R/S		x
	or						
	subtraction		f	PRGM	R/S		x
	or						
	multiplication		GTO	17	R/S		x
	or						
	division		GTO	11	R/S		x
5	For imaginary part		x⇄y				y
6	For next calculation in chain, go						
	to step 3.						
7	For new case, go to step 2.						

Examples:

1. $(1.2 + 3.7i) - (2.6 - 1.9i) = -1.4 + 5.6i$

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

3. $\left[\dfrac{(3 + 4i) + (7.4 - 5.6i)}{(7 - 2i)}\right] [3.1 + 4.6i] = 3.61 + 7.16i$

COMPLEX FUNCTIONS $|z|$, z^2, $^1/z$, $\sqrt{z}$

A complex number $z = a + ib$ has polar representation $re^{i\theta}$. The formulas used to evaluate the given functions are as follows:

1. $|z| = r$

2. $z^2 = r^2 e^{i2\theta}$

3. $^1/z = \frac{1}{r} e^{-i\theta}, z \neq 0$

4. $\sqrt{z} = \pm (\sqrt{r} \ e^{i\theta/2}) = \pm (x + iy)$

The answer is represented by $x + iy$.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00			25	13 00	GTO 00
01	15 09	g →P	26		
02	13 00	GTO 00	27		
03	15 09	g →P	28		
04	15 02	g x^2	29		
05	21	x⇄y	30		
06	31	↑	31		
07	51	+	32		
08	21	x⇄y	33		
09	14 09	f →R	34		
10	13 00	GTO 00	35		
11	15 09	g →P	36		
12	15 22	g 1/x	37		
13	21	x⇄y	38		
14	32	CHS	39		
15	21	x⇄y	40		
16	14 09	f →R	41		
17	13 00	GTO 00	42		
18	15 09	g →P	43		
19	14 02	f $\sqrt{x}$	44		
20	21	x⇄y	45		
21	02	2	46		
22	71	÷	47		
23	21	x⇄y	48		
24	14 09	f →R	49		

REGISTERS
R_0
R_1
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Key in z	b	↑				
		a					
3	For $\|z\|$		f	PRGM	R/S		$\|z\|$
	or						
	z^2		GTO	03	R/S		x
			x⇄y				y
	or						
	$^1/z$		GTO	11	R/S		x
			x⇄y				y
	or						
	$\sqrt{z}$		GTO	18	R/S		x
			x⇄y				y
4	For new case, go to step 2.						

Examples:

1. $|12 - 5i| = 13.00$

2. $(6 - i)^2 = 35.00 - 12.00i$

3. $\dfrac{1}{2 + 5i} = 0.07 - 0.17i$

4. $\sqrt{3 + 4i} = \pm (2.00 + 1.00i)$

DETERMINANT AND INVERSE OF A 2 x 2 MATRIX

Let $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ be a 2 x 2 matrix.

The determinant of A denoted by Det A or |A| is evaluated by the following formula:

$$\text{Det } A = a_{22}\, a_{11} - a_{12}\, a_{21}$$

Also, the program evaluates the multiplicative inverse A^{-1} of A. The following formula is used:

$$A^{-1} = \begin{bmatrix} a_{22}/\text{Det } A & -a_{12}/\text{Det } A \\ -a_{21}/\text{Det } A & a_{11}/\text{Det } A \end{bmatrix}$$

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	24 04	RCL 4
02	24 01	RCL 1
03	61	x
04	24 02	RCL 2
05	24 03	RCL 3
06	61	x
07	41	–
08	23 00	STO 0
09	74	R/S
10	24 04	RCL 4
11	24 00	RCL 0
12	71	÷
13	74	R/S
14	24 02	RCL 2
15	24 00	RCL 0
16	71	÷
17	32	CHS
18	74	R/S
19	24 03	RCL 3
20	24 00	RCL 0
21	71	÷
22	32	CHS
23	74	R/S
24	24 01	RCL 1

DISPLAY		KEY ENTRY
LINE	CODE	
25	24 00	RCL 0
26	71	÷
27	13 00	GTO 00
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 Det A
R_1 a_{11}
R_2 a_{12}
R_3 a_{21}
R_4 a_{22}
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store matrix	a_{11}	STO	1			
		a_{12}	STO	2			
		a_{21}	STO	3			
		a_{22}	STO	4			
3	Compute determinant		f	PRGM	R/S		Det A
4	Compute inverse		R/S				a_{11}^{-1}
			R/S				a_{12}^{-1}
			R/S				a_{21}^{-1}
			R/S				a_{22}^{-1}
5	For new case, go to step 2.						

Example:

Find the determinant and inverse of the matrix

$$A = \begin{bmatrix} 3 & 2 \\ 4 & -4 \end{bmatrix}$$

Solution:

$$\text{Det } A = -20$$

$$A^{-1} = \begin{bmatrix} 0.20 & 0.10 \\ 0.20 & -0.15 \end{bmatrix}$$

NUMBER IN BASE b TO NUMBER IN BASE 10

This program consists of two subprograms. The first changes the integer part of a number in base b to a number in base 10.

$$I_{10} = i_n i_{n-1} \ldots i_2 i_1 = i_n b^{n-1} + i_{n-1}\, b^{n-2} + \ldots + i_2 b + i_1$$

This is evaluated in the form

$$b\,(\ldots (b\,(b\,(i_n\, b + i_{n-1}) + i_{n-2}) + \ldots) + i_2) + i_1$$

The second subprogram changes the fraction part of a number in base b to a number in base 10.

$$F_{10} = f_1 f_2 \ldots f_m = f_1 b^{-1} + f_2 b^{-2} + \ldots + f_m b^{-m}$$

Together the two programs can convert any number in base b to a number in base 10. Zeros must be entered in their proper place.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	23 01	STO 1
02	24 00	RCL 0
03	31	↑
04	31	↑
05	31	↑
06	24 01	RCL 1
07	74	R/S
08	23 01	STO 1
09	34	CLX
10	51	+
11	61	x
12	24 01	RCL 1
13	51	+
14	13 07	GTO 07
15	24 00	RCL 0
16	15 22	g 1/x
17	23 02	STO 2
18	23 03	STO 3
19	61	x
20	74	R/S
21	24 02	RCL 2
22	24 03	RCL 3
23	61	x
24	23 03	STO 3
25	61	x
26	51	+
27	13 20	GTO 20
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 b
R_1 Used
R_2 b^{-1}
R_3 b^{-j}
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store base	b	STO	0			
3	For integer part, input left most						
	digit	i_n	f	PRGM	R/S		
4	Perform for j = n−1,..., 2:						
	Input next digit	i_j*	R/S				
5	Input final digit	i_1*	R/S				I_{10}
6	For fractional part, input digit						
	after decimal	f_1	GTO	15	R/S		
7	Perform for j = 2,..., m−1:						
	Input next digit	f_j*	R/S				
8	Input final digit	f_m*	R/S				F_{10}
9	For new case, go to step 2.						
	* The stack must be maintained						
	at these points.						

Examples:

1. $1777_8 = 1023_{10}$

2. $143.2044_5 = 48.4384_{10}$

NUMBER IN BASE 10 TO NUMBER IN BASE b

This program will convert any positive number in base 10, N_{10}, to a number in base b, N_b, where $2 \leqslant b \leqslant 100$. The algorithm used is an iterative one which adds one more digit to N_b at each iteration. The program pauses as each new N_b is computed to display successive approximations to the final answer. When the displayed value of N_b has reached the accuracy desired by the user, he should press [R/S] to halt the program, then [RCL] [3] to display N_b.

Notes:

1. When the base b is such that $11 \leqslant b \leqslant 100$, two display positions are allocated to each digit of N_b. Begin partitioning to the right and to the left of the decimal point. For example, 41106.12 in base 16 stands for 4B6.C.

2. An error indication during execution means that the machine's accuracy has been exceeded. The value of N_b is in R_3.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 00	RCL 0
02	01	1
03	00	0
04	14 51	f x≥y
05	13 09	GTO 09
06	01	1
07	00	0
08	00	0
09	23 02	STO 2
10	00	0
11	23 03	STO 3
12	24 01	RCL 1
13	14 07	f LN
14	24 00	RCL 0
15	14 07	f LN
16	71	÷
17	15 41	g x<0
18	13 21	GTO 21
19	14 01	f INT
20	13 24	GTO 24
21	14 01	f INT
22	01	1
23	41	–
24	23 04	STO 4

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	24 02	RCL 2
26	21	x⇄y
27	14 03	f y^x
28	24 03	RCL 3
29	51	+
30	23 03	STO 3
31	14 74	f PAUSE
32	14 74	f PAUSE
33	24 00	RCL 0
34	24 04	RCL 4
35	14 03	f y^x
36	23 41 01	STO – 1
37	13 12	GTO 12
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 b
R_1 N_{10}
R_2 10 or 100
R_3 N_b
R_4 1 digit
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Set display format		f	FIX	9		
3	Store base and decimal number	b	STO	0			
		N_{10}	STO	1	f	PRGM	
4	Display successive approximat-						
	ions to N_b		R/S				(N_b)
5	When number is shown with						
	desired accuracy, press [R/S] to						
	halt, then		RCL	3			N_b
6	For new case, go to step 3.						

Examples:

1. $67.32_{10} = 403.050114_{16}$
   $= 43.51E_{16}$

2. $\pi = 3.141592654_{10} = 11.00100100_2$

VECTOR CROSS PRODUCT

If A = (a_1, a_2, a_3) and B = (b_1, b_2, b_3) are two three dimensional vectors then the cross product of A and B is denoted by A x B and is calculated as follows:

$$A \times B = \left(\begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix}, - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix}, \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} \right) = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$$

Let the solution be represented by (c_1, c_2, c_3).

DISPLAY		KEY ENTRY	DISPLAY		KEY ENTRY
LINE	CODE		LINE	CODE	
00			25		
01	24 02	RCL 2	26		
02	24 06	RCL 6	27		
03	61	x	28		
04	24 03	RCL 3	29		
05	24 05	RCL 5	30		
06	61	x	31		
07	41	–	32		
08	74	R/S	33		
09	24 03	RCL 3	34		
10	24 04	RCL 4	35		
11	61	x	36		
12	24 01	RCL 1	37		
13	24 06	RCL 6	38		
14	61	x	39		
15	41	–	40		
16	74	R/S	41		
17	24 01	RCL 1	42		
18	24 05	RCL 5	43		
19	61	x	44		
20	24 02	RCL 2	45		
21	24 04	RCL 4	46		
22	61	x	47		
23	41	–	48		
24	13 00	GTO 00	49		

REGISTERS
R_0
R_1 a_1
R_2 a_2
R_3 a_3
R_4 b_1
R_5 b_2
R_6 b_3
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store A	a_1	STO	1			
		a_2	STO	2			
		a_3	STO	3			
3	Store B	b_1	STO	4			
		b_2	STO	5			
		b_3	STO	6			
4	Compute cross-product		f	PRGM	R/S		c_1
			R/S				c_2
			R/S				c_3
5	For new case, go to step 2.						

Example:

Let A = (2, 5, 2)

B = (3, 3, –4).

Solution:

A x B = (–26, 14, –9)

ANGLE BETWEEN, NORM, AND DOT PRODUCT OF VECTORS

Let $\vec{a} = (a_1, a_2 \ldots, a_n)$ and $\vec{b} = (b_1, b_2, \ldots, b_n)$ be two vectors.

The norm of $\vec{a}$ is denoted by $|\vec{a}|$ and is calculated by the following formula:

$$|\vec{a}| = \sqrt{a_1{}^2 + a_2{}^2 + \ldots + a_n{}^2}$$

similarly,

$$|\vec{b}| = \sqrt{b_1{}^2 + b_2{}^2 + \ldots + b_n{}^2}$$

The dot product of $\vec{a}$ and $\vec{b}$ is denoted by $\vec{a} \cdot \vec{b}$ and is calculated by the following formula:

$$\vec{a} \cdot \vec{b} = a_1\, b_1 + a_2\, b_2 + \ldots + a_n\, b_n$$

The angle between a and b is denoted by θ and is calculated by the following formula:

$$\theta = \cos^{-1}\left(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}\right)$$

The angle is calculated in any angular mode. When calculated in degrees, decimal degrees are assumed.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	31	↑
02	15 02	g x^2
03	23 51 01	STO + 1
04	22	R↓
05	21	x⇄y
06	31	↑
07	15 02	g x^2
08	23 51 00	STO + 0
09	22	R↓
10	61	x
11	23 51 02	STO + 2
12	13 00	GTO 00
13	24 02	RCL 2
14	24 00	RCL 0
15	24 01	RCL 1
16	61	x
17	14 02	f $\sqrt{x}$
18	71	÷
19	15 05	g COS^{-1}
20	13 00	GTO 00
21		
22		
23		
24		

DISPLAY		KEY ENTRY
LINE	CODE	
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 Σa_i^2
R_1 Σb_i^2
R_2 $\Sigma a_i b_i$
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Perform for i = 1,..., n:						
	Key in a_i and b_i	a_i	↑				
		b_i	R/S				
4	Find norm of $\vec{a}$		RCL	0	f	$\sqrt{x}$	$\|\vec{a}\|$
5	Find norm of $\vec{b}$		RCL	1	f	$\sqrt{x}$	$\|\vec{b}\|$
6	Find $\|\vec{a}\cdot\vec{b}\|$		RCL	2			$\|\vec{a}\cdot\vec{b}\|$
7	Compute angle between $\vec{a}$ and $\vec{b}$		GTO	13	R/S		θ

Example:

Let $A = (2, 5, 2)$
$B = (3, 3, -4)$

Solution:

$|\vec{a}| = 5.74$

$|\vec{b}| = 5.83$

$\vec{a} \cdot \vec{b} = 13.00$

$\theta = 67.16^\circ$

SIMULTANEOUS EQUATIONS IN TWO UNKNOWNS

Let $ax + by = e$

and $cx + dy = f$

be a system of two equations in two unknowns. Cramer's Rule is used to find the solution.

$$x = \frac{\begin{vmatrix} e & b \\ f & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{ed - bf}{ad - bc} \qquad y = \frac{\begin{vmatrix} a & e \\ c & f \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{af - ec}{ad - bc}$$

If $ad - bc = 0$ the calculator displays *Error*. In this case no solution or no unique solution exists.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 03	RCL 3
02	24 05	RCL 5
03	61	x
04	24 02	RCL 2
05	24 06	RCL 6
06	61	x
07	41	–
08	24 01	RCL 1
09	24 05	RCL 5
10	61	x
11	24 02	RCL 2
12	24 04	RCL 4
13	61	x
14	41	–
15	23 00	STO 0
16	71	÷
17	74	R/S
18	24 01	RCL 1
19	24 06	RCL 6
20	61	x
21	24 03	RCL 3
22	24 04	RCL 4
23	61	x
24	41	–

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	24 00	RCL 0
26	71	÷
27	13 00	GTO 00
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 ad – bc
R_1 a
R_2 b
R_3 e
R_4 c
R_5 d
R_6 f
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store constants	a	STO	1			
		b	STO	2			
		e	STO	3			
		c	STO	4			
		d	STO	5			
		f	STO	6			
3	Find x and y		f	PRGM	R/S		x
			R/S				y
4	For new case, go to step 2.						

Example:

$5x - 3y = 12$

$2x + y = 9$

Solution:

$x = 3.55$

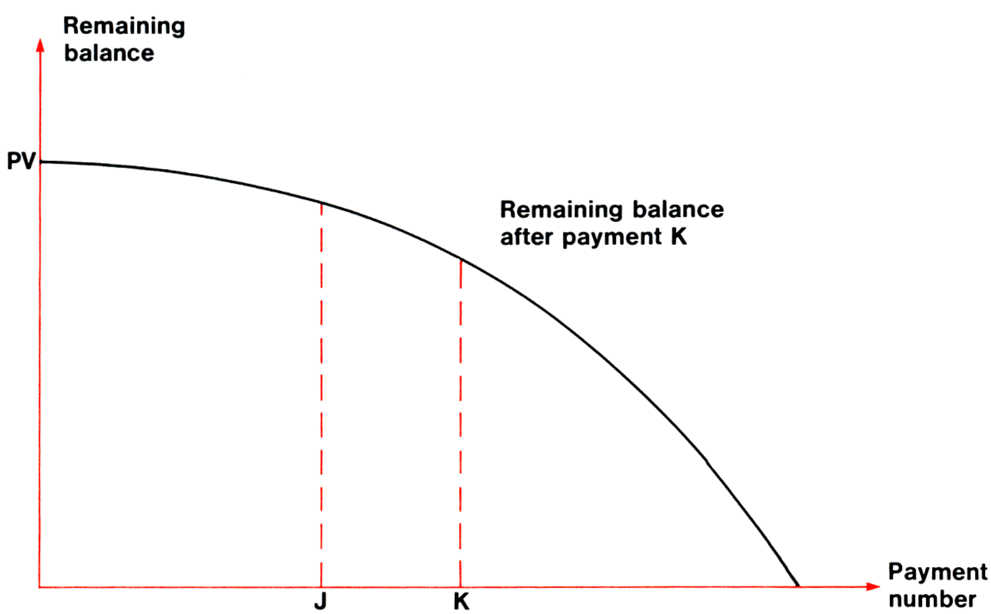
$y = 1.91$

CHAPTER 2 FINANCE

Because many 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.

MORTGAGE LOAN
ACCUMULATED INTEREST/REMAINING BALANCE



As one enters into the realm of financial calculations, one of the most striking revelations is how much of the repayment of a loan goes to interest. A new homeowner, for example, sends off his first monthly installment of $220.13 toward repayment of a 30-year, $30,000 mortgage assumed at 8% annual interest. With a proud sigh and a swelling chest, the homeowner mentally checks $220 off the $30,000 and figures he's well on his way. Right? Well, not quite. In fact, $200 of that payment will go to interest, and only $20.13 to reducing the principal of the loan.

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.

Programming Remarks:

In many finance programs, the expressions (1 + i) and $(1 + i)^n$ are used several times per program. It is often simpler to calculate the quantity once and then store it for later use, rather than calculate it anew each time. In this program, the values of $(1 + i)^{-K}$ and $(1 + i)^{-J}$ are calculated once and then stored in R_7, thus saving both program steps and execution time. The same principle, of course, applies to other expressions in other problems.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	COMMENTS
00							
01	24 01	RCL 1	i				Calculate BAL_K
02	01	1	1	i			
03	51	+	1 + i				
04	24 05	RCL 5	K	1 + i			
05	32	CHS	-K	1 + i			
06	14 03	f y^x	$(1 + i)^{-K}$				
07	23 07	STO 7	$(1 + i)^{-K}$				
08	01	1	1	$(1 + i)^{-K}$			
09	41	–	$(1 + i)^{-K}-1$				
10	24 01	RCL 1	i	$(1 + i)^{-K}-1$			
11	71	÷	s				Let s = $[(1 + i)^{-K}-1] \div i$
12	24 02	RCL 2	PMT	s			
13	61	×	PMT s				
14	24 03	RCL 3	PV	PMT s			
15	51	+	PMT s + PV				
16	24 07	RCL 7	$(1 + i)^{-K}$	PMT s + PV			
17	71	÷	BAL_K				
18	23 06	STO 6	BAL_K				
19	24 01	RCL 1	i	BAL_K			Calculate BAL_{J-1}
20	01	1	1	i	BAL_K		
21	51	+	(1 + i)	BAL_K			
22	24 04	RCL 4	J	(1 + i)	BAL_K		
23	01	1	1	J	(1 + i)	BAL_K	
24	41	–	J – 1	(1 + i)	BAL_K	BAL_K	
25	32	CHS	– (J – 1)	(1 + i)	BAL_K	BAL_K	
26	14 03	f y^x	$(1 + i)^{-(J-1)}$	BAL_K	BAL_K	BAL_K	
27	23 07	STO 7	$(1 + i)^{1-J}$	BAL_K	BAL_K	BAL_K	
28	01	1	1	$(1 + i)^{1-J}$	BAL_K	BAL_K	
29	41	–	$(1 + i)^{1-J}-1$	BAL_K	BAL_K	BAL_K	
30	24 01	RCL 1	i	$(1 + i)^{1-J}-1$	BAL_K	BAL_K	
31	71	÷	s	BAL_K	BAL_K	BAL_K	Let s = $[(1 + i)^{1-J}-1] \div i$
32	24 02	RCL 2	PMT	s	BAL_K	BAL_K	
33	61	×	PMT s	BAL_K	BAL_K	BAL_K	
34	24 03	RCL 3	PV	PMT s	BAL_K	BAL_K	
35	51	+	PMT s + PV	BAL_K	BAL_K	BAL_K	
36	24 07	RCL 7	$(1 + i)^{1-J}$	PMT s + PV	BAL_K	BAL_K	
37	71	÷	BAL_{J-1}	BAL_K	BAL_K	BAL_K	
38	41	–	Diff	BAL_K	BAL_K	BAL_K	Diff = $BAL_K - BAL_{J-1}$
39	24 05	RCL 5	K	Diff	BAL_K	BAL_K	K – J + 1 gives no. PMT's
40	24 04	RCL 4	J	K	Diff	BAL_K	from J through K
41	41	–	K–J	Diff	BAL_K	BAL_K	
42	01	1	1	K–J	Diff	BAL_K	
43	51	+	K – J + 1	Diff	BAL_K	BAL_K	
44	24 02	RCL 2	PMT	m	Diff	BAL_K	m = K – J + 1
45	61	×	m PMT	Diff	BAL_K	BAL_K	m PMT is $ paid, J–K
46	51	+	Int_{J-K}	BAL_K	BAL_K	BAL_K	Display Int_{J-K}
47	74	R/S	Int_{J-K}	BAL_K	BAL_K	BAL_K	
48	21	x⇄y	BAL_K	Int_{J-K}	BAL_K	BAL_K	Display BAL_K
49	13 00	GTO 00	BAL_K	Int_{J-K}	BAL_K	BAL_K	

REGISTERS	
R_0	
R_1	i
R_2	PMT
R_3	PV
R_4	J
R_5	K
R_6	BAL_K
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	f	PRGM	
3	Compute accumulated interest						
	from payments J through K.		R/S				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.						

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? Also, generate a schedule of interest paid and remaining balance for the first 5 years of the mortgage (periods 12, 24, 36, 48, 60).

Solution:

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

.08 [↑] 12 [÷] [STO] [1] 200 [STO] [2] 25000 [STO] [3] 1
[STO] [4] 3 [STO] [5] [f] [PRGM] [R/S] ⟶ 499.33
(interest paid in 1974)

[R/S] ⟶ 24899.33
(remaining balance at end of 1974)

4 [STO] [4] 15 [STO] [5] [R/S] ⟶ 1976.65
(interest paid in 1975)

[R/S] ⟶ 24475.98
(remaining balance at end of 1975)

Now, generate the amortization schedule:

1 STO 4 12 STO 5 R/S → 1985.00
(interest thru 1st year)

R/S → 24585.00
(remaining balance after 1st year)

24 STO 5 R/S → 3935.56
(interest thru 2nd year)

R/S → 24135.56
(remaining balance after 2nd year)

36 STO 5 R/S → 5848.81
(interest thru 3rd year)

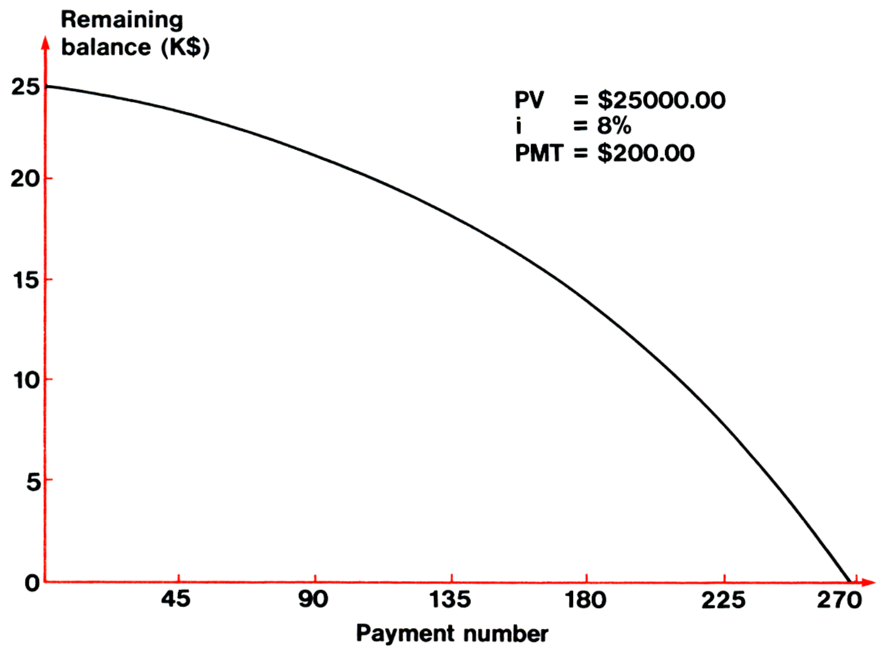
R/S → 23648.81
(remaining balance after 3rd year)

48 STO 5 R/S → 7721.67
(interest thru 4th year)

R/S → 23121.67
(remaining balance after 4th year)

60 STO 5 R/S → 9550.77
(interest thru 5th year)

R/S → 22550.77
(remaining balance after 5th year)



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:

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

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

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	01	1
02	24 02	RCL 2
03	01	1
04	51	+
05	24 01	RCL 1
06	32	CHS
07	14 03	f y^x
08	41	–
09	24 02	RCL 2
10	21	x⇄y
11	71	÷
12	24 04	RCL 4
13	61	x
14	13 00	GTO 00
15	01	1
16	24 02	RCL 2
17	01	1
18	51	+
19	24 01	RCL 1
20	32	CHS
21	14 03	f y^x
22	41	–
23	24 02	RCL 2
24	71	÷

DISPLAY		KEY ENTRY
LINE	CODE	
25	24 03	RCL 3
26	61	x
27	13 00	GTO 00
28	01	1
29	24 04	RCL 4
30	24 03	RCL 3
31	71	÷
32	24 02	RCL 2
33	61	x
34	41	–
35	14 07	f LN
36	24 02	RCL 2
37	01	1
38	51	+
39	14 07	f LN
40	71	÷
41	32	CHS
42	13 00	GTO 00
43		
44		
45		
46		
47		
48		
49		

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 program						
2	For payment	n	STO	1			
		i	STO	2			
		PV	STO	4			
			f	PRGM	R/S		PMT
3	For present value	n	STO	1			
		i	STO	2			
		PMT	STO	3			
			GTO	15	R/S		PV
4	For number of payments	i	STO	2			
		PMT	STO	3			
		PV	STO	4			
			GTO	28	R/S		n
5	For new case, go to step 2, 3, or						
	4.						

Examples:

1. What monthly payment is required to amortize a $3000 loan at 9.5% (.095) in 36 months?

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

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%?

Solutions:

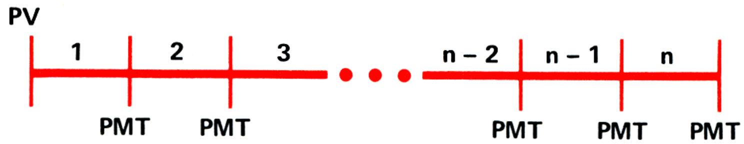
Divide 0.095 by 12 to find the monthly interest rate expressed as a decimal.

1. $96.10

2. $3811.43

3. 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

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

The initial guess for i is given by

$$i_o = \frac{PMT}{PV} - \frac{PV}{n^2\ PMT} .$$

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

DISPLAY		KEY ENTRY
LINE	CODE	
25	15 22	g 1/x
26	01	1
27	51	+
28	71	÷
29	01	1
30	51	+
31	24 05	RCL 5
32	61	x
33	01	1
34	41	–
35	24 02	RCL 2
36	71	÷
37	71	÷
38	23 51 02	STO + 2
39	15 03	g ABS
40	33	EEX
41	06	6
42	32	CHS
43	14 41	f x<y
44	13 10	GTO 10
45	24 02	RCL 2
46	13 00	GTO 00
47		
48		
49		

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

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store number of payments	n	STO	1			
3	Key in present value and pay-						
	ment amount	PV	↑				
		PMT	÷	STO	3		PV/PMT
4	Compute interest		f	PRGM	R/S		i (decimal)
			EEX	2	x		i (%)
5	For new case, go to step 2.						

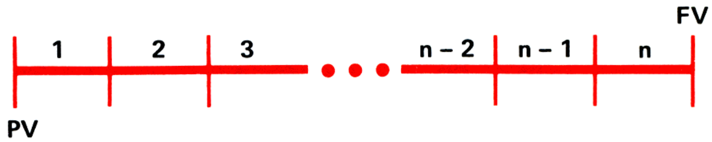
Example:

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

Solution:

15.01%

COMPOUND AMOUNT



This program applies to an amount of principal that has been placed into an account and compounded periodically, with no further deposits. The important variables in this case are the number of compounding periods n, the periodic interest rate i, the principal or present value PV, the future value of the account FV, and the amount of interest accrued I. Any of these may be calculated from the others by these formulas:

$$n = \frac{\ln(FV/PV)}{\ln(1+i)} \qquad i = \left(\frac{FV}{PV}\right)^{1/n} - 1 \qquad PV = FV(1+i)^{-n}$$

$$FV = PV(1+i)^n \qquad I = PV\left[(1+i)^n - 1\right]$$

DISPLAY		KEY
LINE	CODE	ENTRY
00		
01	24 05	RCL 5
02	24 04	RCL 4
03	71	÷
04	14 07	f LN
05	24 02	RCL 2
06	01	1
07	51	+
08	14 07	f LN
09	71	÷
10	13 00	GTO 00
11	24 05	RCL 5
12	24 04	RCL 4
13	71	÷
14	24 01	RCL 1
15	15 22	g 1/x
16	14 03	f y^x
17	01	1
18	41	–
19	13 00	GTO 00
20	24 02	RCL 2
21	01	1
22	51	+
23	24 01	RCL 1
24	32	CHS

DISPLAY		KEY
LINE	CODE	ENTRY
25	14 03	f y^x
26	24 05	RCL 5
27	61	x
28	13 00	GTO 00
29	24 02	RCL 2
30	01	1
31	51	+
32	24 01	RCL 1
33	14 03	f y^x
34	24 04	RCL 4
35	61	x
36	13 00	GTO 00
37	24 02	RCL 2
38	01	1
39	51	+
40	24 01	RCL 1
41	14 03	f y^x
42	01	1
43	41	–
44	24 04	RCL 4
45	61	x
46	13 00	GTO 00
47		
48		
49		

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

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	To compute number of periods	i (decimal)	STO	2			
		PV	STO	4			
		FV	STO	5			
			f	PRGM	R/S		n
3	To compute periodic interest						
	rate	n	STO	1			
		PV	STO	4			
		FV	STO	5			
			GTO	11	R/S		i (decimal)
4	To compute principal	n	STO	1			
		i (decimal)	STO	2			
		FV	STO	5			
			GTO	20	R/S		PV
5	To compute future value	n	STO	1			
		i (decimal)	STO	2			
		PV	STO	4			
			GTO	29	R/S		FV
6	To compute accrued interest	n	STO	1			
		i (decimal)	STO	2			
		PV	STO	4			
			GTO	37	R/S		I
7	For new case, go to step 2, 3, 4,						
	5, or 6.						

Examples:

1. Assuming an annual inflation rate of 10%, how long will it take prices to double? (Suggestion: let PV = 1, FV = 2)

2. Find the rate of return on $1000 compounded quarterly if it amounts to $1500 in 5 years.

3. How much will you need to invest today at 5 3/4% compounded quarterly to have $3000 in 5 years?

4. What is the future value of $2000 invested at 5 3/4% compounded quarterly for 4 years (16 quarters)?

5. How much interest do you receive on $1500 deposited for 10 years if interest at 5 1/2% is compounded annually?

Solutions:

1. 7.27 years
2. .0205 quarterly = 8.19% annually
3. $2255.02 (i = 0.0575/4)
4. $2513.08 (i = 0.0575/4)
5. $1062.22 (i = 0.055)

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:

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

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

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	24 02	RCL 2
02	24 05	RCL 5
03	61	x
04	24 03	RCL 3
05	71	÷
06	24 02	RCL 2
07	01	1
08	51	+
09	23 00	STO 0
10	51	+
11	14 07	f LN
12	24 00	RCL 0
13	14 07	f LN
14	71	÷
15	01	1
16	41	–
17	13 00	GTO 00
18	24 05	RCL 5
19	24 02	RCL 2
20	61	x
21	24 02	RCL 2
22	01	1
23	51	+
24	71	÷

DISPLAY		KEY ENTRY
LINE	CODE	
25	14 73	f LASTx
26	24 01	RCL 1
27	14 03	f y^x
28	01	1
29	41	–
30	71	÷
31	13 00	GTO 00
32	24 03	RCL 3
33	24 02	RCL 2
34	01	1
35	51	+
36	61	x
37	14 73	f LASTx
38	24 01	RCL 1
39	14 03	f y^x
40	01	1
41	41	–
42	61	x
43	24 02	RCL 2
44	71	÷
45	13 00	GTO 00
46		
47		
48		
49		

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 compute number of pay-						
	ments	i (decimal)	STO	2			
		PMT	STO	3			
		FV	STO	5			
			f	PRGM	R/S		n
3	To compute periodic payment						
	amount	n	STO	1			
		i (decimal)	STO	2			
		FV	STO	5			
			GTO	18	R/S		PMT
4	To compute future value	n	STO	1			
		i (decimal)	STO	2			
		PMT	STO	3			
			GTO	32	R/S		FV
5	For new case, go to step 2, 3,						
	or 4.						

Examples:

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

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 6 1/2%?

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.

Solutions:

1. 29.62 quarters or 7.40 years (i = .06/4)

2. \$93.82 (n = 84, i = .065/12)

3. \$5929.92 (n = 36, i = .06/12)

DISCOUNTED CASH FLOW
NET PRESENT VALUE, INTERNAL RATE OF RETURN

The primary purpose of this program is to compute the net present value of a series of cash flows. In general, an initial investment V_0 is made in some enterprise which is expected to bring in periodic cash flows $C_1, C_2, \ldots, C_n$. Given a discount rate i, which must be entered as a decimal, then for each cash flow C_k, the program will compute the net present value at period k, NPV_k. A negative value for NPV_k indicates that the enterprise has not yet been profitable. A positive NPV_k means that the enterprise has been profitable, to the extent that a rate of return i on the original investment has been exceeded.

The program may also be used iteratively to calculate an internal rate of return. The objective here is to find the discount rate i which will make the final net present value, NPV_n, equal to zero. The procedure, then, is to store V_0 and a first guess at the rate of return i, input the cash flows C_1 through C_n; and thus find NPV_n. If NPV_n is negative, the estimated rate of return was too high; if NPV_n is positive, the estimate for i was too low. Adjust the estimate for i accordingly, store the new i, and input the cash flows again. Inspect the new value of NPV_n to obtain a new estimate for i and repeat the process. The entire procedure is repeated until NPV_n is zero, or very close to it. The last value of i input is then regarded as the internal rate of return.

Each figure for net present value is found by

$$NPV_k = -V_0 + \sum_{j=1}^{k} \frac{C_j}{(1+i)^j}$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 01	RCL 1
02	01	1
03	23 04	STO 4
04	51	+
05	23 02	STO 2
06	71	÷
07	24 00	RCL 0
08	41	−
09	24 04	RCL 4
10	14 74	f PAUSE
11	21	x⇄y
12	23 03	STO 3
13	74	R/S
14	24 02	RCL 2
15	24 04	RCL 4
16	01	1
17	51	+
18	23 04	STO 4
19	14 03	f y^x
20	71	÷
21	24 03	RCL 3
22	51	+
23	13 09	GTO 09
24		

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 V_0
R_1 i
R_2 (1 + i)
R_3 NPV_k
R_4 k
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store initial investment and						
	discount rate	V_0	STO	0			
		i (decimal)	STO	1	f	PRGM	
3	Perform for k = 1,..., n:						
	Input C_k and compute NPV_k	C_k	R/S				(k)
							NPV_k
4	For new case, go to step 2.						

Example:

You have been offered an investment opportunity for $150,000 at a capital cost of 10% after taxes. Based on the following cash flows, will this investment be profitable?

Year	Cash Flow
1	$30,000
2	26,300
3	50,000
4	55,600
5	45,200

Solutions:

Remember to enter i as 0.10.

NPV_1 = –$122,727.27

NPV_2 = –$100,991.74

NPV_3 = –$63,426.00

NPV_4 = –$25,450.45

NPV_5 = $2,615.20

Since C_5 is positive the cash flow is profitable to the extent that the cost of capital is 10%.

CALENDAR
DAY OF THE WEEK
DAYS BETWEEN TWO DATES

This program will compute 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 computing 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

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

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, f 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.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	03	3
02	24 01	RCL 1
03	14 41	f x<y
04	13 09	GTO 09
05	01	1
06	51	+
07	24 03	RCL 3
08	13 15	GTO 15
09	01	1
10	03	3
11	51	+
12	24 03	RCL 3
13	01	1
14	41	−
15	03	3
16	06	6
17	05	5
18	73	•
19	02	2
20	05	5
21	61	x
22	14 01	f INT
23	21	x⇄y
24	03	3

DISPLAY		KEY ENTRY
LINE	CODE	
25	00	0
26	73	•
27	06	6
28	61	x
29	14 01	f INT
30	51	+
31	24 02	RCL 2
32	51	+
33	06	6
34	02	2
35	01	1
36	00	0
37	04	4
38	09	9
39	41	−
40	74	R/S
41	07	7
42	71	÷
43	15 01	g FRAC
44	07	7
45	61	x
46	13 00	GTO 00
47		
48		
49		

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	Compute N(m, d, y)		f	PRGM	R/S		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 new case, go to step 2.						
8	For day of week (0 = Sunday)		R/S				Day (0,..., 6)
9	For 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.

Solutions:

1. Thursday (4). (Remember to add 2 days.)
2. 9872.