MATHEMATICS

PROGRAM LIBRARY

sinclair

Enterprise Programmable

QUICK GUIDE TO USING THE LIBRARY

Always refer to the instruction book until you are familiar with the use of the calculator.

To enter a program:

> Enter goto/0/0/prog
> Then enter the keystrokes as given in the table on the right hand side of each program in the library.
> Then enter prog/goto/0/1

Always remember to press [▲] when the upper case of a key is required.

To use a program follow the pre-execution (if applicable) and execution sequence given with it. Remember to wait till the display lights up before entering a number in the middle of an execution sequence.

If you think you have made a mistake in program entry, check the program with some data for which you know the correct answer. If there is an error, either re-enter the program, or find the error using the check codes and correct it as detailed in the instruction book.

If you make a mistake in the execution sequence, it is generally necessary to enter C/goto/0/1 and to start the pre-execution and execution sequences again.

It is a good idea to press [C] to clear any previous results before starting an execution sequence or, indeed, any calculation.

A program can be halted in the middle of execution by entering /stop/ (i.e. pressing [▲] [+/–]).

CONTENTS

17. COMPLEX NUMBERS

COMPLEX MULTIPLICATION

$z = (x + iy)(x' + iy')$

Execution:

x/x◀▶y/y/run/
x'/x◀▶y/y'/run/
real part of z/x◀▶y/imaginary part of z

KEY	#	KEY	#
HALT	00	x◀▶y	40
sto	01	rcl	41
0	02	5	42
x◀▶y	03	goto	43
sto	04	0	44
1	05	0	45
stop	06		46
sto	07		47
2	08		48
x◀▶y	09		49
sto	10		50
3	11		51
×	12		52
rcl	13		53
1	14		54
−	15		55
(	16		56
rcl	17		57
0	18		58
×	19		59
rcl	20		60
2	21		61
)	22		62
=	23		63
sto	24		64
5	25		65
rcl	26		66
3	27		67
×	28		68
rcl	29		69
0	30		70
+	31		71
(	32		72
rcl	33		73
1	34		74
×	35		75
rcl	36		76
2	37		77
)	38		78
=	39		79

COMPLEX DIVISION

KEY	#	KEY	#
HALT	00	0	40
sto	01	×	41
0	02	rcl	42
x◀▶y	03	3	43
sto	04	−	44
1	05	(	45
stop	06	rcl	46
sto	07	2	47
2	08	×	48
x◀▶y	09	rcl	49
sto	10	1	50
3	11	)	51
x^2	12	÷	52
+	13	rcl	53
rcl	14	4	54
2	15	=	55
x^2	16	x◀▶y	56
=	17	rcl	57
sto	18	5	58
4	19	goto	59
rcl	20	0	60
0	21	0	61
×	22		62
rcl	23		63
2	24		64
+	25		65
(	26		66
rcl	27		67
1	28		68
×	29		69
rcl	30		70
3	31		71
)	32		72
÷	33		73
rcl	34		74
4	35		75
=	36		76
sto	37		77
5	38		78
rcl	39		79

$$z = \frac{x + iy}{x' + iy'}$$

Execution:

x/x◀▶y/y/run/
x'/x◀▶y/y'/run/
real part of z/x◀▶y/
imaginary part of z

COMPLEX ARITHMETIC

$$(x + iy) + (x' + iy') = (x + x') + i(y + y')$$

$$(x + iy) - (x' + iy') = (x - x') + i(y - y')$$

$$(x + iy)(x' + iy') = (xx' - yy') + i(xy' + yx')$$

$$\frac{x + iy}{x' + iy'} = \frac{xx' + yy'}{x'^2 + y'^2} + i\frac{yx' - xy'}{x'^2 + y'^2}$$

Execution:

Addition:
x/x◀▶y/y/run/x'/x◀▶y/y'/run/sum (real part)/x◀▶y/imaginary part

Subtraction
x/x◀▶y/y/run/x'/+/−/x◀▶y/y'/+/−/run/real difference x◀▶y/imaginary part

Multiplication:
x/x◀▶y/y/run/x'/x◀▶y/y'/goto/4/3/run/real part of product/x◀▶y/imaginary part

Division:
x/x◀▶y/y/run/x'/x◀▶y/y'/goto/1/9/run/real part of quotient/x◀▶y/imaginary part of quotient

KEY	#	KEY	#
HALT	00	x◀▶y	40
sto	01	rcl	41
0	02	2	42
x◀▶y	03	sto	43
sto	04	2	44
1	05	x◀▶y	45
stop	06	sto	46
M +	07	3	47
0	08	×	48
rcl	09	rcl	49
0	10	1	50
x◀▶y	11	−	51
M +	12	(	52
1	13	rcl	53
rcl	14	0	54
1	15	×	55
goto	16	rcl	56
0	17	2	57
0	18	)	58
sto	19	=	59
2	20	sto	60
x◀▶y	21	5	61
sto	22	rcl	62
3	23	3	63
x^2	24	×	64
+	25	rcl	65
rcl	26	0	66
2	27	+	67
x^2	28	(	68
÷	29	rcl	69
rcl	30	1	70
2	31	×	71
+/−	32	rcl	72
x◀▶y	33	2	73
=	34	)	74
sto	35	=	75
2	36	x◀▶y	76
rcl	37	0	77
3	38	rcl	78
=	39	5	79

KEY	#	KEY	#
HALT	00	=	40
sto	01	D ▶ R	42
0	02	goto	42
x◀▶y	03	0	43
sto	04	0	44
1	05		45
x^2	06		46
+	07		47
rcl	08		48
0	09		49
x^2	10		50
=	11		51
$\sqrt{x}$	12		52
stop	13		53
1/x	14		54
×	15		55
rcl	16		56
1	17		57
=	18		58
arc	19		59
cos	20		60
sto	21		61
1	22		62
rcl	23		63
0	24		64
gin	25		65
3	26		66
4	27		67
rcl	28		68
1	29		69
D ▶ R	30		70
goto	31		71
0	32		72
0	33		73
3	34		74
6	35		75
0	36		76
−	37		77
rcl	38		78
1	39		79

MAGNITUDE AND ARGUMENT OF A COMPLEX NUMBER

$x + iy = re^{i\theta}$

Execution:

x/x◀▶y/y/run/r/run/θ

The value of θ is given in the range $[0, 2\pi)$.

NTH ROOTS OF UNITY

$x_k = e^{2k\pi i/N}$

Execution:

N/run/k/run/ real part of x_k/ x◀▶y/ imaginary part

KEY	#	KEY	#
HALT	00		40
1/x	01		41
×	02		42
stop	03		43
×	04		44
3	05		45
6	06		46
0	07		47
=	08		48
sto	09		49
0	10		50
sin	11		51
x◀▶y	12		52
rcl	13		53
0	14		54
cos	15		55
goto	16		56
0	17		57
0	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

NTH ROOTS OF -1

KEY	#	KEY	#
HALT	00		40
1/x	01		41
×	02		42
(	03		43
stop	04		44
×	05		45
2	06		46
+	07		47
1	08		48
)	09		49
×	10		50
1	11		51
8	12		52
0	13		53
=	14		54
sto	15		55
0	16		56
sin	17		57
x◀▶y	18		58
rcl	19		59
0	20		60
cos	21		61
goto	22		62
0	23		63
0	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$$x_k = e^{(2k+1)\pi i/N}$$

Execution:

N/run/k/run/real part of x_k/
x◀▶y/imaginary part

COMPLEX NUMBER RAISED TO A REAL POWER

$z = (x + iy)^t$

Execution:

x/x◀▶y/y/run/|x + iy|/t/run/ real part of z/x◀▶y/imaginary part of z

Error will be returned if t is too large. Only one value is returned for z — that corresponding to an argument of t times the principle argument of x + iy.

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	1	41
0	02	×	42
x◀▶y	03	rcl	43
sto	04	3	44
1	05	=	45
x^2	06	sto	46
+	07	1	47
rcl	08	cos	48
0	09	×	49
x^2	10	(	50
=	11	rcl	51
$\sqrt{x}$	12	2	52
sto	13	y^x	53
2	14	rcl	54
stop	15	3	55
sto	16	=	56
3	17	sto	57
rcl	18	3	58
1	19	)	59
÷	20	=	60
rcl	21	sto	61
2	22	2	62
=	23	rcl	63
arc	24	1	64
cos	25	sin	65
sto	26	×	66
1	27	rcl	67
rcl	28	3	68
0	29	=	69
gin	30	x◀▶y	70
3	31	rcl	71
4	32	2	72
goto	33	goto	73
4	34	0	74
0	35	0	75
3	36		76
6	37		77
0	38		78
—	39		79

SINE AND COSINE OF A COMPLEX NUMBER

KEY	#	KEY	#
HALT	00		40
sto	01		41
1	02		42
x◀▶y	03	sto	43
arc	04	1	44
D▶R	05	x◀▶y	45
sto	06	arc	46
0	07	D▶R	47
rcl	08	sto	48
1	09	0	49
e^x	10	rcl	50
+	11	1	51
1/x	12	e^x	52
÷	13	+	53
2	14	1/x	54
×	15	÷	55
rcl	16	2	56
0	17	×	57
sin	18	rcl	58
=	19	0	59
sto	20	cos	60
2	21	=	61
rcl	22	sto	62
1	23	2	63
e^x	24	rcl	64
−	25	1	65
1/x	26	e^x	66
÷	27	−	67
2	28	1/x	68
×	29	÷	69
rcl	30	2	70
0	31	×	71
cos	32	rcl	72
=	33	0	73
x◀▶y	34	sin	74
rcl	35	=	75
2	36	+/−	76
goto	37	x◀▶y	77
0	38	rcl	78
0	39	2	79

$$\sin(x + iy) = \sin x \cosh y + i \cos x \sinh y$$

$$\cos(x + iy) = \cos x \cosh y - i \sin x \sinh y$$

Pre-execution:

sin: goto/0/1
cos: goto/4/3

Execution:

x/x◀▶y/y/run/real part/x◀▶y/imaginary part

18. SUMS AND MEANS

SUM OF SQUARES

finds

$$\sum_{i=1}^{n} x_i^2$$

Execution:

x_1/run/x_2/run/
x_3/run/. . . x_n/run/

At each step the sum so far is displayed

Before using the calculator for another purpose: clear.

KEY	#	KEY	#
HALT	00		40
x^2	01		41
+	02		42
goto	03		43
0	04		44
0	05		45
	06		46
	07		47
	08		48
	09		49
	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

SUM OF PRODUCTS (E.G. VECTOR DOT PRODUCT)

KEY	#	KEY	#
HALT	00		40
×	01		41
stop	02		42
+	03		43
goto	04		44
0	05		45
0	06		46
	07		47
	08		48
	09		49
	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

finds

$$\sum_{i=1}^{n} x_i y_i$$

Execution:

x_1/run/y_1/run/
x_2/run/y_2/run/
. . ./x_n/run/y_n/

run/ $\sum_{i=1}^{n} x_i y_i$

After each y_i is entered, the sum so far,

$\sum_{j=1}^{i} x_j y_j$, is displayed.

Remember to clear the calculator after using this program.

HARMONIC SUM (E.G. LENSES IN SERIES)

$$\frac{1}{x} = \frac{1}{x_1} + \ldots + \frac{1}{x_n}$$

Execution:

x_1/run/x_2/run/. . .
. . . x_n/run/x

At each step the harmonic sum so far is displayed.

To start again:
goto/0/1/

Remember to clear the calculator before using it for something else.

KEY	#	KEY	#
HALT	00		40
1/x	01		41
+	02		42
(	03		43
1/x	04		44
stop	05		45
1/x	06		46
)	07		47
goto	08		48
0	09		49
2	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

KEY	#	KEY	#
HALT	00		40
x^2	01		41
+	02		42
(	03		43
$\sqrt{x}$	04		44
stop	05		45
x^2	06		46
)	07		47
goto	08		48
0	09		49
2	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

PYTHAGOREAN SUM (AS IN THE THEOREM OF THAT NAME)

$$x = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

Execution:

x_1/run/x_2/run/
. . . x_n/run/x.

At each stage the Pythagorean sum so far is displayed.

To start again go to /0/1/

Remember to clear the calculator before using it for something else.

SUM OF SERIES $F(x_i)$

Finds

$$\sum_{i=1}^{n} f(x_i)$$

Write a program segment to evaluate $f(x_i)$ where x_i is on display. Insert it in the program opposite.

Execution:

0/run/x_1/run/x_2/run/

. . . x_n/run/ $\sum_{i=1}^{n} f(x_i)$

At each stage sum so far is displayed.

Remember in writing your program not to use more than one level of parenthesis, as one level is already used.

KEY	#	KEY	#
HALT	00		40
+	01		41
(	02		42
stop	03		43
YOUR SEGMENT	04		44
	05		45
	06		46
	07		47
	08		48
	09		49
	10		50
	11		51
	12		52
	13		53
)	14		54
=	15		55
goto	16		56
0	17		57
1	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

KEY	#	KEY	#
HALT	00		40
+	01		41
(	02		42
stop	03		43
sto	04		44
0	05		45
+	06		46
rcl	07		47
0	08		48
1/x	09		49
=	10		50
arc	11		51
D ▶ R	12		52
cos	13		53
+	14		54
rcl	15		55
0	16		56
=	17		57
1/x	18		58
)	19		59
=	20		60
goto	21		61
0	22		62
1	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

Example:

To find

$$\sum_{i=1}^{n}\left(\frac{1}{x_i + \cos\left(x_i + \frac{1}{x_i}\right)}\right)$$

(x_i in radians)

$$f(x) = \frac{1}{x_i + \cos\left(x_i + \frac{1}{x_i}\right)}$$

so a suitable program segment is

sto/0/+/rcl/0/

$\frac{1}{x}$/=/arc/D ▶ R/cos/

+/rcl/0/=/$\frac{1}{x}$/

The complete program is as opposite.

0/run/3/run/5/
run/7/run/9/
run/ 9.3324-01

So $\dfrac{1}{3 + \cos(3 + \frac{1}{3})}$

$+ \dfrac{1}{5 + \cos(5 + \frac{1}{5})}$

$+ \dfrac{1}{7 + \cos(7 + \frac{1}{7})}$

$+ \dfrac{1}{9 + \cos(9 + \frac{1}{9})}$

$= .93324$

ARITHMETIC MEAN

Execution:

x_1/run/x_2/run/

. . . /x_n/run/Arithmetic mean

At each stage the arithmetic mean so far is displayed

To start again:
goto/0/1/

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
1	03		43
sto	04		44
1	05		45
rcl	06		46
0	07		47
stop	08		48
M +	09		49
0	10		50
1	11		51
M +	12		52
1	13		53
rcl	14		54
0	15		55
÷	16		56
rcl	17		57
1	18		58
=	19		59
goto	20		60
0	21		61
8	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

WEIGHTED MEAN

KEY	#	KEY	#
HALT	00		40
×	01		41
stop	02		42
=	03		43
sto	04		44
0	05		45
1	06		46
sto	07		47
1	08		48
stop	09		49
×	10		50
stop	11		51
=	12		52
M +	13		53
0	14		54
1	15		55
M +	16		56
1	17		57
rcl	18		58
0	19		59
÷	20		60
rcl	21		61
1	22		62
=	23		63
goto	24		64
0	25		65
9	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$$W = \frac{W_1X_1 + W_1X_2 + \ldots + W_nX_n}{n}$$

Execution:

W_1/run/X_1/run/
W_2/run/X_2/run/
. . . W_n/run/X_n/
run/ W

At each stage (after entering x_i)W so far is displayed.

To start again:
goto/0/1/

WEIGHTED MEAN (SEVEN OR FEWER ENTRIES)

$$W = \frac{W_1X_1 + W_2X_2 + \ldots + W_7X_7}{7}$$

This program can be adapted to fewer than seven entries by omitting the parts referring to the unused memories.

It can be used with fewer than seven entries without modification if

goto/0/1/

is used prior to each execution.

Pre-execution:

Store W_1 to W_7 in memories 0 to 6 respectively. The program may now be used many times without re-entering the weights.

Execution:

X_1/run/X_2/run/X_3/run/. . .

. . ./X_7/run/W

After each entry the weighted mean so far is displayed.

KEY	#	KEY	#
HALT	00	)	40
×	01	+	41
rcl	02	(	42
0	03	÷	43
+	04	5	44
(	05	=	45
stop	06	stop	46
×	07	×	47
rcl	08	rcl	48
1	09	5	49
)	10	)	50
+	11	+	51
(	12	(	52
÷	13	÷	53
2	14	6	54
=	15	=	55
stop	16	stop	56
×	17	×	57
rcl	18	rcl	58
2	19	6	59
)	20	)	60
+	21	÷	61
(	22	7	62
÷	23	=	63
3	24	goto	64
=	25	0	65
stop	26	0	66
×	27		67
rcl	28		68
3	29		69
)	30		70
+	31		71
(	32		72
÷	33		73
4	34		74
=	35		75
stop	36		76
×	37		77
rcl	38		78
4	39		79

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
1	03		43
sto	04		44
1	05		45
rcl	06		46
0	07		47
stop	08		48
×	09		49
rcl	10		50
0	11		51
=	12		52
sto	13		53
0	14		54
1	15		55
M +	16		56
1	17		57
rcl	18		58
0	19		59
y^x	20		60
rcl	21		61
1	22		62
1/x	23		63
=	24		64
goto	25		65
0	26		66
8	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

GEOMETRIC MEAN

Execution:

x_1/run/x_2/run/
. . . x_n/run/

Geometric mean

At each stage the geometric mean so far is displayed.

To start again
goto/0/1/

HARMONIC MEAN

$$\frac{1}{H} = \frac{1}{n}\left(\frac{1}{x_1} + \ldots + \frac{1}{x_n}\right)$$

Execution:

x_1/run/x_2/run/
. . . x_n/run/H

At each stage the harmonic mean so far is displayed.

To start again:
goto/0/1/

KEY	#	KEY	#
HALT	00		40
1/x	01		41
sto	02		42
0	03		43
1	04		44
sto	05		45
1	06		46
rcl	07		47
0	08		48
1/x	09		49
stop	10		50
1/x	11		51
M +	12		52
0	13		53
1	14		54
M +	15		55
1	16		56
rcl	17		57
0	18		58
÷	19		59
rcl	20		60
1	21		61
=	22		62
1/x	23		63
goto	24		64
1	25		65
0	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

ROOT MEAN SQUARE

KEY	#	KEY	#
HALT	00		40
x^2	01		41
sto	02		42
0	03		43
1	04		44
sto	05		45
1	06		46
rcl	07		47
0	08		48
$\sqrt{x}$	09		49
stop	10		50
x^2	11		51
M +	12		52
0	13		53
1	14		54
M +	15		55
1	16		56
rcl	17		57
0	18		58
$\sqrt{x}$	19		59
÷	20		60
rcl	21		61
1	22		62
=	23		63
goto	24		64
1	25		65
0	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$$R = \sqrt{\frac{x_1^2 + x_2^2 + \ldots + x_n^2}{n}}$$

Execution:

x_1/run/x_2/run
. . . x_n/run/R

At each stage the r.m.s. so far is displayed.

To start again:
goto/0/1/

GENERAL MEAN

$$g(n) = \frac{\sum_{i=1}^{n} f(x_i, i)}{n}$$

Write a program segment to find $f(x_i, i)$ where x_i is in memory 0 and i is in memory 1. (The principle here is the same as in the program to sum the series $f(x_i)$.)

Execution:

0/run/x_1/run/x_2/run/. . ./x_n/run/$g(n)$

After each entry g(i) so far is displayed

To start again:
goto/0/1/

KEY	#	KEY	#
HALT	00	goto	40
sto	01	0	41
2	02	5	42
sto	03		43
1	04		44
stop	05		45
sto	06		46
0	07		47
1	08		48
M +	09		49
1	10		50
↑	11		51
	12		52
	13		53
	14		54
YOUR SEGMENT	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
↓	31		71
M +	32		72
2	33		73
rcl	34		74
2	35		75
÷	36		76
rcl	37		77
1	38		78
=	39		79

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
×	03		43
stop	04		44
=	05		45
sto	06		46
1	07		47
÷	08		48
rcl	09		49
0	10		50
=	11		51
stop	12		52
M +	13		53
0	14		54
×	15		55
stop	16		56
=	17		57
M +	18		58
1	19		59
rcl	20		60
1	21		61
goto	22		62
0	23		63
8	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

NORMALISED SUM OF PRODUCTS

$$S = \frac{\Sigma x_i y_i}{\Sigma x_i}$$

Execution:

x_1/run/y_1/
run/x_2/run/y_2/run/
. . ./x_n/run/y_n/run/ S

At each stage the normalised sum so far is displayed after y_i/run/.

To start again:
goto/0/1/.

If the x_i are always the same and you do not wish to re-enter them for each execution and there are seven or less of them, use program 8 omitting ÷/7/= at step 61.

$W_i = X_i / \sum_{i=1}^{n} X_i$.

19. EQUATION SOLVING

19.1 Quadratic equations

19.2 Simultaneous equations in two variables

To solve simultaneous equations in three variables use Cramer's rule:

if
$$a_1x + b_1y + c_1z = d_1$$
$$a_2x + b_2y + c_2z = d_2$$
$$a_3x + b_3y + c_3z = d_3$$

then
$$x = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix} \div \Delta$$

$$y = \begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix} \div \Delta$$

$$z = \begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix} \div \Delta$$

where $\Delta = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$

Use the program in Section 26 for 3×3 determinants

For simultaneous equations in more variables Cramer's rule can still be used and the determinant program is still helpful.

19.3 Cubic equations – by the Newton Raphson Method

19.4 Finds the reduced cubic

19.5 Finds the roots of a reduced cubic in the case in which there are three real unequal roots – we leave the general implementation of Cardan's method as a challenge to the reader.

19.6 Interval halving

19.7 Newton-Raphson method

19.8 Secant method

We recommend method 6. as the best way to solve all equations in one variable apart from quadratics

QUADRATIC EQUATIONS

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

Execution:

a/run/b/run/c/
run/x^+/run/x^-
where $x^\pm$ are the roots

$$x^\pm = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

If Error is displayed the equation has complex roots
$x \pm iy$

use the following sequence:
C/goto/4/0/run
real part, x/run/
imaginary part, y

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	1	41
0	02	stop	42
stop	03	rcl	43
÷	04	0	44
2	05	+/−	45
÷	06	$\sqrt{x}$	46
rcl	07	=	47
0	08	goto	48
+/−	09	0	49
=	10	0	50
sto	11		51
1	12		52
x^2	13		53
−	14		54
(	15		55
stop	16		56
÷	17		57
rcl	18		58
0	19		59
)	20		60
=	21		61
sto	22		62
0	23		63
$\sqrt{x}$	24		64
+	25		65
rcl	26		66
1	27		67
=	28		68
stop	29		69
rcl	30		70
1	31		71
−	32		72
rcl	33		73
0	34		74
$\sqrt{x}$	35		75
=	36		76
goto	37		77
0	38		78
0	39		79

SIMULTANEOUS EQUATIONS IN TWO VARIABLES

$a_1 x + b_1 y = c_1$

$a_2 x + b_2 y = c_2$

Execution:

a_1/run/b_1/run/c_1/run/a_2/run/b_2/run/c_2/run/x/run/y

If Error is indicated there is no unique solution.

KEY	#	KEY	#
HALT	00	(	40
sto	01	rcl	41
0	02	1	42
stop	03	×	43
sto	04	rcl	44
1	05	5	45
stop	06	)	46
sto	07	÷	47
2	08	rcl	48
stop	09	6	49
sto	10	=	50
3	11	stop	51
stop	12	rcl	52
sto	13	0	53
4	14	×	54
stop	15	rcl	55
sto	16	5	56
5	17	—	57
rcl	18	(	58
0	19	rcl	59
×	20	3	60
rcl	21	×	61
4	22	rcl	62
—	23	2	63
(	24	)	64
rcl	25	÷	65
3	26	rcl	66
×	27	6	67
rcl	28	=	68
1	29	goto	69
)	30	0	70
=	31	0	71
sto	32		72
6	33		73
rcl	34		74
2	35		75
×	36		76
rcl	37		77
4	38		78
—	39		79

SOLUTION OF A CUBIC BY THE NEWTON RAPHSON METHOD

$f(x) = ax^3 + bx^2 + cx + d = 0$

Successive approximations x_i to a root of f are found by

$$x_{i+1} = \frac{2ax_i^3 + bx_i^2 - d}{3ax_i^2 + 2bx_i + c}$$

Choose x_0

Pre-execution:

goto/0/1

Execution:

a/run/b/run/c/run/
d/run/x_0/run/x_1
/run/x_2
/run/x_3
/run/x_4 etc

If the x_i fail to converge try a different x_0. Remember that there are three distinct real roots only if $\Delta < 0$ where Δ is defined in program 5.

KEY	#	KEY	#
HALT	00	(	40
sto	01	rcl	41
0	02	1	42
stop	03	×	43
sto	04	2	44
1	05	)	45
stop	06	×	46
sto	07	rcl	47
2	08	4	48
stop	09	+	49
sto	10	rcl	50
3	11	2	51
stop	12	)	52
sto	13	=	53
4	14	goto	54
×	15	1	55
2	16	2	56
×	17		57
rcl	18		58
0	19		59
+	20		60
rcl	21		61
1	22		62
×	23		63
rcl	24		64
4	25		65
x^2	26		66
−	27		67
rcl	28		68
3	29		69
÷	30		70
(	31		71
rcl	32		72
4	33		73
×	34		74
3	35		75
×	36		76
rcl	37		77
0	38		78
+	39		79

REDUCED CUBIC

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	2	41
0	02	)	42
stop	03	÷	43
÷	04	9	44
rcl	05	+	45
0	06	rcl	46
=	07	0	47
sto	08	=	48
1	09	goto	49
stop	10	0	50
=	11	0	51
sto	12		52
2	13		53
stop	14		54
=	15		55
sto	16		56
0	17		57
rcl	18		58
1	19		59
x^2	20		60
÷	21		61
3	22		62
+/−	23		63
+	24		64
rcl	25		65
2	26		66
=	27		67
stop	28		68
×	29		69
rcl	30		70
1	31		71
+/−	32		72
×	33		73
2	34		74
−	35		75
(	36		76
rcl	37		77
1	38		78
×	39		79

Substituting

$$y = x + \frac{b}{3a}$$

into

$$ax^3 + bx^2 + cx + d = 0$$

we obtain

$$y^3 + py + q = 0$$

Execution:

a/run/b/run/c/run
d/run/p/run/q

CARDAN'S METHOD

Solution of

$y^3 + py + q = 0$

in the case

$\Delta = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2 < 0$

i.e. three unequal real roots.

The roots are

$y_1 = u + v$

$y_2 = -\frac{u+v}{2} + \frac{u-v}{2} i\sqrt{3}$

$y_3 = -\frac{u+v}{2} - \frac{u-v}{2} i\sqrt{3}$

where

$u = \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}}$

$v = \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}}$

Execution:

p/run/q/run/y_1/run/y_2/run/y_3

KEY	#	KEY	#
HALT	00	0	40
x^2	01	=	41
$\sqrt{x}$	02	cos	42
÷	03	×	43
3	04	rcl	44
=	05	0	45
$\sqrt{x}$	06	×	46
sto	07	2	47
0	08	+/−	48
x^2	09	=	49
×	10	stop	50
rcl	11	rcl	51
0	12	1	52
=	13	−	53
1/x	14	6	54
×	15	0	55
stop	16	=	56
÷	17	cos	57
2	18	×	58
+/−	19	rcl	59
=	20	0	60
arc	21	×	61
cos	22	2	62
÷	23	+/−	63
3	24	=	64
=	25	goto	65
sto	26	0	66
1	27	0	67
cos	28		68
×	29		69
rcl	30		70
0	31		71
×	32		72
2	33		73
=	34		74
stop	35		75
rcl	36		76
1	37		77
+	38		78
6	39		79

EQUATION SOLVING BY INTERVAL HALVING

KEY	#	KEY	#
HALT	00		40
sto	01		41
2	02		42
stop	03		43
sto	04		44
1	05	gin	45
rcl	06	1	46
2	07	3	47
sto	08	+/−	48
0	09	gin	39
goto	10	0	50
1	11	6	51
7	12	rcl	52
rcl	13	2	53
2	14	stop	54
sto	15	goto	55
1	16	0	56
rcl	17	1	57
0	18		58
+	19		59
rcl	20		60
1	21		61
÷	22		62
2	23		63
=	24		64
sto	25		65
2	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
f(rcl2)	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

(to 7 sig fig)

To solve

$f(x) = 0$

write a program segment to evaluate $f(x)$ when x is in memory 2.

Choose x and y such that

$f(x) > 0, \quad f(y) < 0$

Execution:

x/run/y/run/x_o

where $f(x_o) = 0$.

f should be continuous in the range concerned — we recommend drawing a sketch graph of f first.

Find all the roots by using different starting values x and y.

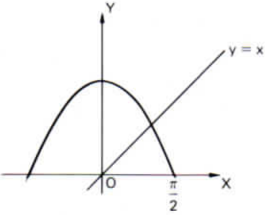
This program may take several minutes to execute.

Example:

To solve

$\cos x = x$

sketch graph:



There is a solution between 0 and $\frac{\pi}{2}$

$f(x) = \cos x - x$

$f(x) > 0,\ f\left(\frac{\pi}{2}\right) < 0$

Our program segment is

rcl/2/arc/D ▶ R/cos/−/rcl/2/=

Execution:

0/run/π/÷/2/=/run/

7.3908 − 01

To recover last three digits:

×/10/=/

7.3908513

So answer is

.73908513 radians.

Note that the accuracy here is limited by the accuracy of the cosine function.

EQUATION SOLVING BY NEWTON RAPHSON METHOD (REQUIRES KNOWLEDGE OF CALCULUS)

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
↑	03		43
	04		44
	05		45
	06		46
	07		47
	08		48
	09		49
YOUR	10		50
	11		51
	12		52
	13		53
SEGMENT	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
↓	27		67
+/−	28		68
+	29		69
rcl	30		70
0	31		71
=	32		72
goto	33		73
0	34		74
0	35		75
	36		76
	37		77
	38		78
	39		79

To solve $f(x) = 0$.
Draw a sketch graph as usual.
Write a program segment to evaluate $f(x)/f'(x)$ (x in memory 0), and insert in the program opposite. Let x_0 be your first approximation.

Execution:

x_0/run/ x_1
/run/ x_2
/run/ x_3
. . .

where x_i are the convergents to the root nearest x_0. If x_i fails to converge try a different x_0.

SECANT METHOD

$f(x) = 0$

Draw a sketch graph of f and find an approximation x_1 to a root of $f(x) = 0$

Find an approximation, k, to the slope of f at x_1 — e.g.

$$k = \frac{f(\eta) - f(\xi)}{\eta - \xi}$$

where η is just a bit more than x_1, and ξ a bit less.

Write a program segment to evaluate f(x) where x is in memory 1 and insert it in the program given here.

Execution:

k/run/x_1/run/x_2/run/x_3/run/x_4/run . . .

where $x_2, x_3, x_4, \ldots$ are successive approximations to a root of $f(x) = 0$, given by the iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{k}$$

If the x_i are converging too slowly decrease k by:

goto/0/1/new k/run/x_n/run . . . etc.

If the x_i are diverging increase k in the same way.

By varying the starting value x_1 to values near other roots (look at your sketch graph), the more precise values of these roots can be found as well.

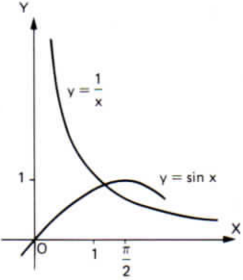
KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
	06		46
	07		47
	08		48
f(x)	09		49
	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
÷	16		56
rcl	17		57
0	18		58
+/−	19		59
+	20		60
rcl	21		61
1	22		62
=	23		63
goto	24		64
0	25		65
3	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
arc	06		46
D ▶ R	07		47
sin	08		48
—	09		49
rcl	10		50
1	11		51
1/x	12		52
=	13		53
÷	14		54
rcl	15		55
0	16		56
+/−	17		57
+	18		58
rcl	19		59
1	20		60
=	21		61
goto	22		62
0	23		63
3	24		64
	25		
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

Example:

To solve $\sin x = \frac{1}{x}$

Sketch Graph



Take $x_1 = 1$

$f(x) = \sin x - \frac{1}{x}$

k = slope of $\sin x$ — slope of $\frac{1}{x}$

$= \text{about } \frac{1}{2} - (-1)$

$= \frac{3}{2}$

Program segment is

arc/D ▶ R/sin/—/rcl/1/$^1/_x$/=

Program is as opposite

Execution:

1.5/run/

1/run/ 1.105686

/run/ 1.1127819

/run/ 1.1139262

/run/ 1.1141182

/run/ 1.1141506

/run/ 1.114156

/run/ 1.1141569

/run/ 1.1141571

/run/ 1.1141571

So answer is

1.1141571

20. SERIES AND PROGRESSIONS

SUMS OF POWERS OF THE NATURAL NUMBERS

$$1 + 2 + \ldots + n = \tfrac{1}{2} n(n+1)$$

$$\sum_{i=1}^{n} i^2 = \tfrac{1}{6} n(n+1)(2n+1)$$

$$\sum_{i=1}^{n} i^3 = \tfrac{1}{4} n^2 (n+1)^2$$

The program has been written in such a way that the program segments corresponding to the above sums may be extracted from the program and used on their own.

Execution:

(a) goto/0/1/n/run/

$\sum_{i=1}^{n} i$

(b) goto/1/5/n/run/

$\sum_{i=1}^{n} i^2$

(c) goto/4/0/n/run/

$\sum_{i=1}^{n} i^3$

KEY	#	KEY	#
HALT	00	×	40
×	01	(	41
(	02	+	42
+	03	1	43
1	04	)	44
)	05	=	45
÷	06	x^2	46
2	07	÷	47
=	08	4	48
stop	09	=	49
goto	10	stop	50
0	11	goto	51
1	12	4	52
	13	0	53
	14		54
×	15		55
(	16		56
+	17		57
1	18		58
=	19		59
×	20		60
(	21		61
×	22		62
2	23		63
—	24		64
1	25		65
)	26		66
)	27		67
÷	28		68
6	29		69
=	30		70
stop	31		71
goto	32		72
1	33		73
5	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

SUMS OF POWERS OF NATURAL NUMBERS

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
×	03		43
(	04		44
+	05		45
1	06		46
)	07		47
÷	08		48
2	09		49
=	10		50
stop	11		51
×	12		52
(	13		53
x^2	14		54
sto	15		55
1	16		56
rcl	17		57
0	18		58
×	19		59
2	20		60
+	21		61
1	22		62
)	23		63
÷	24		64
3	25		65
=	26		66
stop	27		67
rcl	28		68
1	29		69
goto	30		70
0	31		71
0	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

Execution:

n/run/ $\sum_{i=1}^{n} i$/run/

$\sum_{i=1}^{n} i^2$/run/ $\sum_{i=1}^{n} i^3$

ARITHMETIC PROGRESSION

a, a + d, a + 2d, . . .

n^{th} term, $\ell = a + (n-1)d$

sum to n terms,

$$s = \frac{n(a+\ell)}{2}$$

common difference,

$$d = \frac{\ell - a}{n-1}$$

Execution:

(a) a/run/n/run/d/
run/ℓ/run/s

OR

(b) a/run/n/run/ℓ
goto/3/1/run/d/run/s

KEY	#	KEY	#
HALT	00	)	40
sto	01	=	41
0	02	stop	42
stop	03	×	43
sto	04	(	44
1	05	rcl	45
stop	06	1	46
×	07	−	47
(	08	1	48
rcl	09	)	49
1	10	÷	50
−	11	2	51
1	12	+	52
)	13	rcl	53
+	14	0	54
rcl	15	×	55
0	16	rcl	56
=	17	1	57
stop	18	=	58
+	19	goto	59
rcl	20	0	60
0	21	0	61
×	22		62
rcl	23		63
1	24		64
÷	25		65
2	26		66
=	27		67
goto	28		68
0	29		69
0	30		70
−	31		71
rcl	32		72
0	33		73
÷	34		74
(	35		75
rcl	36		76
1	37		77
−	38		78
1	39		79

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
y^x	06		46
(	07		47
stop	08		48
−	09		49
1	10		50
)	11		51
×	12		52
rcl	13		53
0	14		54
=	15		55
stop	16		56
×	17		57
rcl	18		58
1	19		59
−	20		60
rcl	21		61
0	22		62
÷	23		63
(	24		64
rcl	25		65
1	26		66
−	27		67
1	28		68
)	29		69
=	30		70
goto	31		71
0	32		72
0	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

GEOMETRIC PROGRESSION

$$S = a + ar + \ldots + ar^{n-1}$$
$$= \frac{a(1-r^n)}{1-r}$$

Execution:

a/run/r/run/n/
run/ar^{n-1}/run/
$\frac{a(1-r^n)}{1-r}$

Range: $r > 0$, $r \neq 1$.

INFINITE GEOMETRIC SERIES

$$S = \sum_{i=0}^{\infty} ar^i = \frac{a}{1-r}$$

Providing $|r| < 1$

Execution:

a/run/r/run/S

KEY	#	KEY	#
HALT	00		40
÷	01		41
(	02		42
1	03		43
−	04		44
stop	05		45
)	06		46
=	07		47
goto	08		48
0	09		49
0	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

HARMONIC SERIES

KEY	#	KEY	#
HALT	00	goto	40
sto	01	0	41
3	02	0	42
stop	03		43
sto	04		44
4	05		45
stop	06		46
sto	07		47
1	08		48
0	09		49
sto	10		50
2	11		51
sto	12		52
0	13		53
rcl	14		54
0	15		55
×	16		56
rcl	17		57
4	18		58
+	19		59
rcl	20		60
3	21		61
=	22		62
1/x	23		63
M +	24		64
2	25		65
1	26		66
M +	27		67
0	28		68
rcl	29		69
0	30		70
−	31		71
rcl	32		72
1	33		73
=	34		74
gin	35		75
1	36		76
4	37		77
rcl	38		78
2	39		79

$$S = \frac{1}{a} + \frac{1}{a+d} + \frac{1}{a+2d} + \ldots + \frac{1}{a+(n-1)d}$$

Execution:

a/run/d/run/n/
run/S

Restriction:
For no $m \leqslant n$ may $a + (m-1)d = 0$

ARITHMETICO-GEOMETRIC SERIES

$a, (a+d)r, (a+2d)r^2, \ldots, (a+(n-1)d)r^{n-1}$

Sum to n terms = S

$$= \frac{ndr^n}{r-1} + \frac{1-r^n}{1-r}\left(a + \frac{dr}{1-r}\right)$$

Execution:

a/run/d/run/r/
run/n/run/S

Restriction:
$r > 0$
$r \neq 1$

KEY	#	KEY	#
HALT	00	rcl	40
+	01	0	41
(	02	)	42
stop	03	÷	43
sto	04	(	44
0	05	rcl	45
×	06	1	46
stop	07	−	47
sto	08	1	48
1	09	)	49
÷	10	=	50
(	11	goto	51
1	12	0	52
−	13	0	53
rcl	14		54
1	15		55
)	16		56
)	17		57
×	18		58
(	19		59
rcl	20		60
1	21		61
y^x	22		62
stop	23		63
sto	24		64
2	25		65
−	26		66
1	27		67
)	28		68
+	29		69
(	30		70
rcl	31		71
1	32		72
y^x	33		73
rcl	34		74
2	35		75
×	36		76
rcl	37		77
2	38		78
×	39		79

INFINITE ARITH-METICO-GEOMETRIC SERIES

KEY	#	KEY	#
HALT	00		40
+	01		41
(	02		42
stop	03		43
×	04		44
stop	05		45
sto	06		46
0	07		47
÷	08		48
(	09		49
1	10		50
−	11		51
rcl	12		52
0	13		53
)	14		54
)	15		55
÷	16		56
(	17		57
1	18		58
−	19		59
rcl	20		60
0	21		61
)	22		62
=	23		63
stop	24		64
goto	25		65
0	26		66
1	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

Sum to infinity,

$$S = \frac{a + \frac{dr}{1-r}}{1-r}$$

$|r| < 1$

Execution:

a/run/d/run/r/run/S

SUMMING A GENERAL SERIES

Write a program segment to evaluate f(n) where n is in memory 0, and insert the segment in the program opposite. The program will then find

$$\sum_{n=a}^{b} f(n)$$

Execution:

a/run/b/run/ $\sum_{a}^{b} f(n)$

Example:

$$\sum_{n=3}^{7} \cos\left(1-\frac{1}{n}\right)\pi$$

$\left(1-\frac{1}{n}\right)\pi$ radians is

$180\left(1-\frac{1}{n}\right)$ degrees so a suitable

program segment is

rcl/0/$\frac{1}{x}$/+/−/+/1/×/1/8/0/=/cos

The program is then entered as on the back of this page.

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
0	03		43
sto	04		44
2	05		45
stop	06		46
sto	07	M +	47
1	08	2	48
	09	rcl	49
	10	0	50
	11	+	51
	12	1	52
	13	=	53
	14	sto	54
	15	0	55
	16	−	56
	17	rcl	57
SPACE	18	1	58
	19	−	59
	20	1	60
	21	=	61
FOR	22	gin	62
	23	0	63
	24	9	64
	25	rcl	65
YOUR	26	2	66
	27	goto	67
f(n)	28	0	68
	29	0	69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

KEY	#	KEY	#
HALT	00	2	40
sto	01	goto	41
0	02	0	42
0	03	0	43
sto	04		44
2	05		45
stop	06		46
sto	07		47
1	08		48
rcl	09		49
0	10		50
1/x	11		51
+/−	12		52
+	13		53
1	14		54
×	15		55
1	16		56
8	17		57
0	18		58
=	19		59
cos	20		60
M +	21		61
2	22		62
rcl	23		63
0	24		64
+	25		65
1	26		66
=	27		67
sto	28		68
0	29		69
−	30		70
rcl	31		71
1	32		72
−	33		73
1	34		74
=	35		75
gin	36		76
0	37		77
9	38		78
rcl	39		79

Example of run

3/run/7/run/−3.783118

so $\sum_{3}^{7} \cos\left(1 - \frac{1}{n}\right)\pi =$

-3.783

SUMMING A GENERAL SERIES TO INFINITY

Write a program segment to evaluate f(n) where n is in memory 0. This program finds

$$\sum_{n=0}^{\infty} f(n)$$

Execution:
goto/2/9/run/

$$\sum_{0}^{\infty} f(n)$$

This sum is given to about four decimal places unless the series converges very slowly. If the series diverges Error is generally indicated, but may not be.

The sum calculated is actually to N terms: N may be recovered from memory 0.

KEY	#	KEY	#
HALT	00	Y	40
+	01	O	41
(	02	U	42
x^2	03	R	43
—	04		44
1	05	S	45
./EE	06	E	46
./EE	07	G	47
9	08	M	48
+/—	09	E	49
=	10	N	50
gin	11	T	51
2	12		52
6	13		53
rcl	14		54
2	15	goto	55
)	16	0	56
=	17	1	57
sto	18		58
2	19		59
1	20		60
M +	21		61
0	22		62
goto	23		63
3	24		64
4	25		65
rcl	26		66
2	27		67
stop	28		68
0	29		69
sto	30		70
2	31		71
sto	32		72
0	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

21. SCIENTIFIC FUNCTIONS

21.1 sinh, cosh, tanh.
21.2 $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$
21.3 sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$

The relevant program segments may be extracted from the programs above and rearranged, so one need only have in a program those hyperbolic functions one wishes to use.

To find sech, cosech, coth note that

$$\operatorname{sech} x = \frac{1}{\cosh x}$$

$$\operatorname{cosech} x = \frac{1}{\sinh x}$$

$$\coth x = \frac{1}{\tanh x} \text{ also}$$

$$\operatorname{sech}^{-1} x = \cosh^{-1}\left(\frac{1}{x}\right) \text{ etc.}$$

HYPERBOLIC FUNCTIONS

Execution:

(a) goto/0/1/x/run/sinh x
range: |x| < 200

(b) goto/1/0/x/run/cosh x
range: |x| < 200

(c) goto/1/9/x/run/tanh x
range: unlimited

KEY	#	KEY	#
HALT	00	e^x	40
e^x	01	−	41
−	02	1	42
(	03	÷	43
1/x	04	(	44
)	05	+	45
÷	06	2	46
2	07	)	47
=	08	=	48
stop	09	stop	49
e^x	10		50
+	11		51
(	12		52
1/x	13		53
)	14		54
÷	15		55
2	16		56
=	17		57
stop	18		58
gin	19		59
3	20		60
7	21		61
×	22		62
2	23		63
+/−	24		64
=	25		65
e^x	26		66
−	27		67
1	28		68
÷	29		69
(	30		70
+/−	31		71
−	32		72
2	33		73
)	34		74
=	35		75
stop	36		76
×	37		77
2	38		78
=	39		79

KEY	#	KEY	#
HALT	00		40
+	01		41
(	02		42
x^2	03		43
+	04		44
1	05		45
=	06		46
$\sqrt{x}$	07		47
)	08		48
=	09		49
ln	10		50
stop	11		51
+	12		52
(	13		53
x^2	14		54
−	15		55
1	16		56
=	17		57
$\sqrt{x}$	18		58
)	19		59
=	20		60
ln	21		61
stop	22		62
+	23		63
1	24		64
÷	25		65
(	26		66
+/−	27		67
+	28		68
2	29		69
)	30		70
=	31		71
$\sqrt{x}$	32		72
ln	33		73
stop	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

INVERSE HYPERBOLICS

Execution:

(a) goto/0/1/x/run/ $\sinh^{-1} x$
range: unlimited

(b) goto/1/2/x/run/ $\cosh^{-1} x$
range: $x \geqslant 1$

(c) goto/2/3/x/run/ $\tanh^{-1} x$
range: $-1 < x < 1$

HYPERBOLICS AND INVERSES

Execution:

(a) goto/0/1/x/run/$\sinh x$
range: $|x| < 200$

(b) goto/1/0/x/run/$\cosh x$
range: $|x| < 200$

(c) goto/1/9/x/run/$\tanh x$
range: $x < 100$

(d) goto/6/6/x/run/$\tanh x$
range: $x > -100$

(e) goto/3/2/x/run/$\sinh^{-1} x$
range: unlimited

(f) goto/4/3/x/run/$\cosh^{-1} x$
range: $x \geqslant 1$

(g) goto/5/4/x/run/$\tanh^{-1} x$
range: $|x| < 1$

KEY	#	KEY	#
HALT	00	=	40
e^x	01	ln	41
−	02	stop	42
(	03	+	43
1/x	04	(	44
)	05	x^2	45
÷	06	−	46
2	07	1	47
=	08	=	48
stop	09	$\sqrt{x}$	49
e^x	10	)	50
+	11	=	51
(	12	ln	52
1/x	13	stop	53
)	14	+	54
÷	15	1	55
2	16	÷	56
=	17	(	57
stop	18	+/−	58
×	19	+	59
2	20	2	60
=	21	)	61
e^x	22	=	62
−	23	$\sqrt{x}$	63
1	24	ln	64
÷	25	stop	65
(	26	×	66
+	27	2	67
2	28	+/−	68
)	29	=	69
=	30	e^x	70
stop	31	−	71
+	32	1	72
(	33	÷	73
x^2	34	(	74
+	35	+/−	75
1	36	−	76
=	37	2	77
$\sqrt{x}$	38	)	78
)	39	=	79

LOG $_a$x

KEY	#	KEY	#
HALT	00		40
log	01		41
1/x	02		42
×	03		43
stop	04		44
log	05		45
=	06		46
goto	07		47
0	08		48
0	09		49
	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

Execution:

a/run/x/run/

$\log_a x$

LOG$_a$x-BASE a TO BE USED REPEATEDLY

Execution:

(a) To enter base: goto/0/9/a/run

(b) To find logarithm: x/run/$\log_a x$

KEY	#	KEY	#
HALT	00		40
log	01		41
÷	02		42
rcl	03		43
0	04		44
=	05		45
goto	06		46
0	07		47
0	08		48
log	09		49
sto	10		50
0	11		51
goto	12		52
0	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

BESSEL FUNCTIONS

KEY	#	KEY	#
HALT	00	sto	40
÷	01	4	41
2	02	1	42
y^x	03	M +	43
(	04	4	44
x^2	05	rcl	45
+/−	06	1	46
sto	07	M +	47
2	08	3	48
rcl	09	÷	49
0	10	rcl	50
)	11	4	51
=	12	÷	52
sto	13	(	53
1	14	rcl	54
0	15	4	55
sto	16	+	56
3	17	rcl	57
1	18	0	58
M +	19	)	59
3	20	×	60
rcl	21	rcl	61
1	22	2	62
÷	23	=	63
rcl	24	sto	64
3	25	1	65
=	26	x^2	66
sto	27	+/−	67
1	28	+	68
x◀▶y	29	1	69
−	30	./EE	70
rcl	31	./EE	71
0	32	9	72
=	33	+/−	73
gin	34	=	74
1	35	gin	75
8	36	4	76
0	37	2	77
sto	38	rcl	78
3	39	3	79

To find J_n (x) to 4 significant figures, n an integer $\geqslant 0$

Pre-execution:

Store n in memory 0.

Execution:

x/run/J_n (x)

Long execution times can be expected for large values of x and n.

22. AREAS AND VOLUMES

CIRCLE

Circumference
$C = 2\pi r$

Area
$A = \pi r^2$

Radius = r

Execution:

(a) goto/0/1/r/run/C

(b) goto/1/0/r/run/A

(c) goto/1/8/C/run/r

(d) goto/2/7/C/run/A

(e) goto/3/8/A/run/r

(f) goto/4/6/A/run/C

It is not necessary if you want to repeat the use of program (a) to start again with
goto/0/1/,

to repeat program (b), you need not re-enter
goto/1/0/,

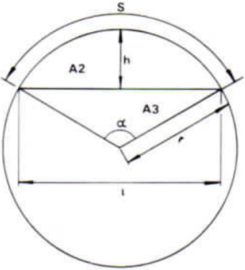
similarly for the other program segments.

The program segments on the right can be used on their own – it is not necessary to enter the entire program – only those parts which are required.

KEY	#	KEY	#
HALT	00	=	40
×	01	$\sqrt{x}$	41
2	02	stop	42
×	03	goto	43
π	04	3	44
=	05	8	45
stop	06	×	46
goto	07	4	47
0	08	×	48
1	09	π	49
x^2	10	=	50
×	11	$\sqrt{x}$	51
π	12	stop	52
=	13	goto	53
stop	14	4	54
goto	15	6	55
1	16		56
0	17		57
÷	18		58
2	19		59
÷	20		60
π	21		61
=	22		62
stop	23		63
goto	24		64
1	25		65
8	26		66
x^2	27		67
÷	28		68
4	29		69
÷	30		70
π	31		71
=	32		72
stop	33		73
goto	34		74
2	35		75
7	36		76
	37		77
÷	38		78
π	39		79

SEGMENT SECTOR AND CHORD OF A CIRCLE

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	0	41
0	02	÷	42
stop	03	2	43
sto	04	=	44
1	05	stop	45
÷	06	−	46
2	07	(	47
=	08	rcl	48
sin	09	0	49
×	10	x^2	50
rcl	11	×	51
0	12	rcl	52
×	13	1	53
2	14	sin	54
=	15	÷	55
stop	16	2	56
rcl	17	)	57
1	18	stop	58
÷	19	=	59
2	20	goto	60
=	21	0	61
cos	22	0	62
+/−	23		63
+	24		64
1	25		65
×	26		66
rcl	27		67
0	28		68
=	29		69
stop	30		70
rcl	31		71
1	32		72
D ▶ R	33		73
×	34		74
rcl	35		75
0	36		76
=	37		77
stop	38		78
×	39		79



Length of chord,

$$\ell = 2r \sin\frac{\alpha}{2}$$

Height of segment,

$$h = r - r\cos\frac{\alpha}{2}$$

Length of arc,

$$S = r\alpha$$

Area of sector,

$$A_1 = A_2 + A_3 = \frac{1}{2} r^2 \alpha$$

Area of segment,

$$A_2 = \frac{1}{2} r^2 \sin\alpha$$

Execution:

r/run/α/run/ℓ/run/
h/run/S/run/A_1/
run/A_2/run/A_3

Enter α in degrees.

N.B. The formulae for ℓ, h, and A_2 are valid when α is measured in degrees *or* radians. The formulae for S and A_1 are valid *only* when α is measured in radians.

The program takes this into account.

SPHERE

Volume,
$V = \frac{4}{3}\pi r^3$

Surface Area,
$A = 4\pi r^2$
r = radius

Execution:

(a) goto/0/1/
r/run/A/run/V

(b) goto/2/0/
V/run/r/run/A

(c) goto/4/4/A/
run/r/run/V

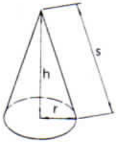
It is not necessary to use the goto /// at the beginning of the execution sequence to repeat the same sort of calculation.

The program segments corresponding to (a), (b) and (c) can, of course, be removed or used on their own. It is not necessary to enter the whole program.

KEY	#	KEY	#
HALT	00	stop	40
x^2	01	goto	41
sto	02	2	42
0	03	0	43
×	04	÷	44
π	05	4	45
×	06	÷	46
4	07	π	47
=	08	=	48
stop	09	$\sqrt{x}$	49
×	10	stop	50
rcl	11	y^x	51
0	12	3	52
−	13	×	53
3	14	π	54
=	15	×	55
stop	16	4	56
goto	17	÷	57
0	18	3	58
1	19	=	59
×	20	stop	60
3	21	goto	61
÷	22	4	62
4	23	4	63
÷	24		64
π	25		65
y^x	26		66
(	27		67
1	28		68
÷	29		69
3	30		70
)	31		71
=	32		72
stop	33		73
x^2	34		74
×	35		75
π	36		76
×	37		77
4	38		78
=	39		79

RIGHT CIRCULAR CONE

KEY	#	KEY	#
HALT	00	=	40
sto	01	goto	41
0	02	0	42
x^2	03	0	43
+	04		44
stop	05		45
sto	06		46
1	07		47
x^2	08		48
=	09		49
$\sqrt{x}$	10		50
stop	11		51
×	12		52
π	13		53
×	14		54
rcl	15		55
0	16		56
=	17		57
stop	18		58
+	19		59
(	20		60
rcl	21		61
0	22		62
x^2	23		63
×	24		64
π	25		65
)	26		66
stop	27		67
=	28		68
stop	29		69
rcl	30		70
0	31		71
x^2	32		72
×	33		73
rcl	34		74
1	35		75
×	36		76
π	37		77
÷	38		78
3	39		79



Slant height

$$s = \sqrt{r^2 + h^2}$$

Curved surface area

$$S = \pi r\sqrt{r^2 + h^2}$$

Base area

$$B = \pi r^2$$

Total surface area

$$A = S + B$$

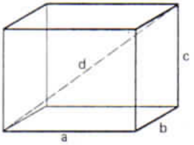
Volume

$$V = \frac{\pi r^2 h}{3}$$

Execution:

r/run/h/run/s/run/
S/run/B/run/A/
run/V

CUBOID



Diagonal

$d = \sqrt{a^2 + b^2 + c^2}$

Surface area

$A = 2(ab + ac + bc)$

Volume

$V = abc$

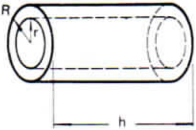
Execution:

a/run/b/run/c/run/
d/run/A/run/V

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	1	41
0	02	×	42
x^2	03	rcl	43
+	04	2	44
stop	05	=	45
sto	06	goto	46
1	07	0	47
x^2	08	0	48
+	09		49
stop	10		50
sto	11		51
2	12		52
x^2	13		53
=	14		54
$\sqrt{x}$	15		55
stop	16		56
rcl	17		57
1	18		58
+	19		59
rcl	20		60
2	21		61
×	22		62
rcl	23		63
0	24		64
+	25		65
(	26		66
rcl	27		67
1	28		68
×	29		69
rcl	30		70
2	31		71
)	32		72
×	33		73
2	34		74
=	35		75
stop	36		76
rcl	37		77
0	38		78
×	39		79

KEY	#	KEY	#
HALT	00	÷	40
sto	01	rcl	41
0	02	2	42
stop	03	=	43
sto	04	stop	44
1	05	×	45
stop	06	2	46
sto	07	+	47
2	08	(	48
rcl	09	rcl	49
0	10	0	50
x^2	11	+	51
×	12	rcl	52
rcl	13	1	53
2	14	×	54
×	15	rcl	55
π	16	2	56
=	17	×	57
stop	18	π	58
÷	19	×	59
rcl	20	2	60
0	21	)	61
×	22	stop	62
2	23	=	63
=	24	goto	64
stop	25	0	65
rcl	26	0	66
0	27		67
x^2	28		68
—	29		69
rcl	30		70
1	31		71
x^2	32		72
×	33		73
rcl	34		74
2	35		75
×	36		76
π	37		77
=	38		78
stop	39		79

CYLINDRICAL TUBE



Volume of outer cylinder
$V_1 = \pi R^2 h$

Curved surface area of outer cylinder
$A_1 = 2\pi Rh$

Volume of Tube
$V_2 = \pi h(R^2 - r^2)$

Area of one anular end of tube
$A_2 = \pi(R^2 - r^2)$

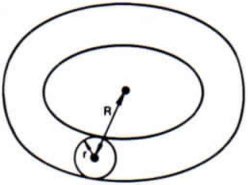
Total curved surface area
$A_3 = 2\pi h(R + r)$

Total surface area
$A_4 = 2A_2 + A_3$

Execution:

R/run/r/
run/h/run/V_1/run/
A_1/run/V_2/run/A_2/
run/A_3/run/A_4

TORUS



Area
$A = 4\pi^2 Rr$

Volume
$V = 2\pi^2 Rr^2$

Execution:

r/run/R/run/A/run/V

KEY	#	KEY	#
HALT	00		40
×	01		41
(	02		42
×	03		43
stop	04		44
×	05		45
π	06		46
x^2	07		47
×	08		48
4	09		49
)	10		50
stop	11		51
÷	12		52
2	13		53
=	14		54
goto	15		55
0	16		56
0	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

23. NUMBER THEORY

FACTORIALS

Execution:

n/run/n!
$n \geqslant 1$.

Error will be displayed if $n! \geqslant 10^{100}$.

In such cases use Stirling's formula.

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
sto	03		43
1	04		44
rcl	05		45
0	06		46
−	07		47
2	08		48
=	09		49
gin	10		50
2	11		51
7	12		52
+	13		53
1	14		54
=	15		55
sto	16		56
0	17		57
×	18		58
rcl	19		59
1	20		60
=	21		61
sto	22		62
1	23		63
goto	24		64
0	25		65
5	26		66
rcl	27		67
1	28		68
goto	29		69
0	30		70
0	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

STIRLING'S FORMULA

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
+	03		43
./EE	04		44
5	05		45
×	06		46
rcl	07		47
0	08		48
ln	09		49
−	10		50
rcl	11		51
0	12		52
+	13		53
./EE	14		54
9	15		55
1	16		56
8	17		57
9	18		58
3	19		59
8	20		60
5	21		61
3	22		62
=	23		63
goto	24		64
0	25		65
0	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

For large n,

$n! \simeq \sqrt{2\pi n}\, e^{-n} n^{n}$.

The program finds ln (n!).

Execution:

n/run/ ln (n!)

BINOMIAL CO-EFFICIENTS

$$\binom{n}{r} = \frac{n!}{r!\,(n-r)!}$$

$n > r > 0$

Execution:

n/run/r/run/ $\binom{n}{r}$

Long execution times are possible with large values of r.

KEY	#	KEY	#
HALT	00	goto	40
sto	01	0	41
0	02	9	42
stop	03	rcl	43
sto	04	2	44
1	05	goto	45
1	06	0	46
sto	07	0	47
2	08		48
×	09		49
rcl	10		50
0	11		51
÷	12		52
rcl	13		53
1	14		54
=	15		55
sto	16		56
2	17		57
rcl	18		58
1	19		59
−	20		60
2	21		61
=	22		62
gin	23		63
4	24		64
3	25		65
+	26		66
1	27		67
=	28		68
sto	29		69
1	30		70
rcl	31		71
0	32		72
−	33		73
1	34		74
=	35		75
sto	36		76
0	37		77
rcl	38		78
2	39		79

HIGHEST COMMON FACTOR AND LEAST COMMON MULTIPLE

KEY	#	KEY	#
HALT	00	sto	40
sto	01	0	41
0	02	goto	42
×	03	1	43
stop	04	0	44
sto	05		45
1	06		46
=	07		47
sto	08		48
2	09		49
rcl	10		50
0	11		51
−	12		52
rcl	13		53
1	14		54
=	15		55
gin	16		56
3	17		57
3	18		58
+/−	19		59
gin	20		60
3	21		61
9	22		62
rcl	23		63
2	24		64
÷	25		65
rcl	26		66
0	27		67
stop	28		68
=	29		69
goto	30		70
0	31		71
0	32		72
+/−	33		73
sto	34		74
1	35		75
goto	36		76
1	37		77
0	38		78
+/−	39		79

Execution:

a/run/b/run/

hcf(a, b)/run/

lcm(a, b)

a, b > 0

PRIME NUMBER TESTER

This program finds the least prime number which divides the given number n. If the least prime divisor is n itself then n is prime.

(Exception: if n = 1, the program will return 1, which is not prime.)

Execution:

n/run/least prime divisor of n

Example:

171/run/3
179/run/179

So 171 is not prime – it is divisible by 3.
179, however, is prime – there is no smaller number which divides it, apart from 1.

Warning:

With large numbers long execution times are possible.

KEY	#	KEY	#
HALT	00	+	40
sto	01	2	41
0	02	=	42
$\sqrt{x}$	03	goto	43
sto	04	0	44
2	05	0	45
1	06	2	46
sto	07	M+	47
1	08	1	48
2	09	rcl	49
÷	10	2	50
rcl	11	−	51
0	12	rcl	52
x◀▶y	13	1	53
−	14	=	54
(	15	gin	55
+	16	6	56
1	17	2	57
./EE	18	x◀▶y	58
./EE	19	goto	59
9	20	1	60
−	21	0	61
1	22	rcl	62
./EE	23	0	63
./EE	24	goto	64
9	25	0	65
)	26	0	66
=	27	2	67
+/−	28	goto	68
gin	29	0	69
4	30	0	70
6	31		71
rcl	32		72
1	33		73
−	34		74
2	35		75
=	36		76
gin	37		77
6	38		78
7	39		79

PRIME FACTOR-ISATION

KEY	#	KEY	#
HALT	00	=	40
sto	01	gin	41
0	02	7	42
÷	03	6	43
2	04	rcl	44
+	05	3	45
1	06	stop	46
./EE	07	rcl	47
./EE	08	1	48
9	09	goto	49
−	10	0	50
1	11	1	51
./EE	12	1	52
./EE	13	+/−	53
9	14	M+	54
=	15	1	55
sto	16	rcl	56
1	17	3	57
sto	18	+/−	58
2	19	M+	59
1	20	2	60
sto	21	goto	61
3	22	2	62
rcl	23	3	63
0	24	1	64
−	25	M+	65
rcl	26	3	66
2	27	rcl	67
=	28	1	68
gin	29	M+	69
5	30	2	70
2	31	goto	71
+/−	32	2	72
gin	33	3	73
6	34	0	74
4	35	0	75
rcl	36	rcl	76
1	37	3	77
−	38	stop	78
2	39	1	79

This program finds the prime factors of a number n. After all the prime factors have been found, 1 is displayed. $n \geqslant 1$.

Execution:

n/run/1st prime factor
/run/2nd prime factor
/run/ . . .
. . .
/run/last prime factor
/run/1

Example:

12/run/2/run/2/run/3/run/1/
So $12 = 2 \times 2 \times 3$

Warning:

Long execution times can be expected for certain large numbers.

CONVERSION FROM BASE m TO DECIMAL

Converts integer whose base m digits are

$d_n d_{n-1} \ldots d_2 d_1$ to decimal equivalent, x.

Execution:

m/run/d_n/run/
d_{n-1}/run/ . . .
d_2/run/d_1/goto/
1/1/run/x

Example:

To convert the base 9 number 518 to decimal:
9/run/5/1/
run/8/goto/1/1/
run/422
So the decimal representation is 422

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
x	04		44
rcl	05		45
0	06		46
+	07		47
goto	08		48
0	09		49
3	10		50
=	11		51
goto	12		52
0	13		53
0	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

KEY	#	KEY	#
HALT	00	1	40
sto	01	+/−	41
0	02	M+	42
sto	03	2	43
1	04	rcl	44
stop	05	2	45
sto	06	gin	46
2	07	5	47
−	08	5	48
rcl	09	1	49
1	10	M+	50
=	11	3	51
gin	12	goto	52
2	13	3	53
8	14	9	54
rcl	15	rcl	55
1	16	3	56
×	17	stop	57
rcl	18	rcl	58
0	19	1	59
=	20	M+	60
sto	21	2	61
1	22	−	62
rcl	23	2	63
2	24	=	64
goto	25	gin	65
0	26	7	66
8	27	1	67
rcl	28	goto	68
1	29	2	69
÷	30	8	70
rcl	31	8	71
0	32	8	72
=	33	8	73
sto	34	8	74
1	35	8	75
0	36	8	76
sto	37	8	77
3	38	8	78
rcl	39	0	79

CONVERSION FROM DECIMAL TO BASE m

Converts decimal integer x to base m representation

$d_n d_{n-1} \ldots d_2 d_1$.

Execution:

m/run/x/run/d_n/run/
d_{n-1}/run . . . /d_1/run/
88888888

The display of 8's indicates the conversion is complete.

Example:

To convert 422 to base 9:
9/run/422/run/5
/run/1/run/8/run/
88888888

So the base 9 representation is 518.

NUMBER BASE CONVERSION SAME BASE USED REPEATEDLY

Store base m in memory 0
Converts decimal integer x to base m representation

$d_n d_{n-1} \ldots d_2 d_1$ and vice versa.

Decimal to base m:

Pre-execution:

goto/1/6/

Execution:

x/run/d_n/
run/d_{n-1}/run/. . . /d_1/
run/88888888
It is not necessary to repeat the pre-execution.

Base m to decimal:

Pre-execution:

goto/0/1

Execution:

d_n/run/d_{n-1}/
run/d_2/run/d_1/=/x
Note the = sign after d_1.

KEY	#	KEY	#
HALT	00	0	40
×	01	sto	41
rcl	02	3	42
0	03	rcl	43
+	04	2	44
goto	05	−	45
0	06	rcl	46
0	07	1	47
8	08	=	48
8	09	gin	49
8	10	6	50
8	11	0	51
8	12	sto	52
8	13	2	53
8	14	1	54
stop	15	M+	55
sto	16	3	56
2	17	goto	57
1	18	4	58
sto	19	3	59
1	20	rcl	60
rcl	21	3	61
1	22	stop	62
×	23	rcl	63
rcl	24	1	64
0	25	÷	65
−	26	rcl	66
rcl	27	0	67
2	28	=	68
x◀▶y	29	sto	69
=	30	1	70
gin	31	−	71
4	32	1	72
0	33	=	73
x◀▶y	34	gin	74
sto	35	0	75
1	36	8	76
goto	37	goto	77
2	38	4	78
1	39	0	79

OCTAL / DECIMAL CONVERSION

KEY	#	KEY	#
HALT	00	1	40
sto	01	—	41
1	02	1	42
0	03	=	43
sto	04	gin	44
2	05	7	45
1	06	8	46
sto	07	rcl	47
5	08	5	48
rcl	09	×	49
1	10	rcl	50
÷	11	3	51
rcl	12	=	52
0	13	goto	53
—	14	0	54
(	15	7	55
+	16	sto	56
1	17	1	57
./EE	18	1	58
./EE	19	0	59
9	20	sto	60
—	21	0	61
1	22	8	62
./EE	23	sto	63
./EE	24	3	64
9	25	goto	65
=	26	0	66
sto	27	3	67
1	28	sto	68
)	29	1	69
×	30	8	70
rcl	31	sto	71
0	32	0	72
×	33	1	73
rcl	34	0	74
5	35	goto	75
=	36	6	76
M+	37	3	77
2	38	rcl	78
rcl	39	2	79

Converts octal integer x into decimal integer y and vice versa.

Although faster than the previous number base programs, may still take fairly long for large numbers.

Octal to Decimal:

Pre-execution:

goto/5/6

Execution:

x/run/y

Decimal to Octal:

Pre-execution:

goto/6/8

Execution:

y/run/x

It is not necessary to repeat the pre-execution sequences to continue with conversions of the same type.

HEXADECIMAL/ DECIMAL CONVERSION

This program is fairly fast. To convert decimal integer x to hex integer $d_n d_{n-1} \ldots d_2 d_1$.

Execution:

(a) hex to decimal:
goto/0/1/d_n/run/
d_{n-1}/run/ . . . /d_2/run/
d_1/=/x

(b) decimal to hex:
goto/0/8
/x/run/d_0/run/d_1/
run/d_2/run/d_3/ . . .
(eventually $d_n = 0$ for all large enough n).

Notice the digits appear in the reverse order in (b).

KEY	#	KEY	#
HALT	00		40
×	01		41
1	02		42
6	03		43
+	04		44
goto	05		45
0	06		46
0	07		47
sto	08		48
0	09		49
rcl	10		50
0	11		51
÷	12		52
1	13		53
6	14		54
−	15		55
(	16		56
+	17		57
1	18		58
./EE	19		59
./EE	20		60
9	21		61
−	22		62
1	23		63
./EE	24		64
./EE	25		65
9	26		66
=	27		67
sto	28		68
0	29		69
)	30		70
×	31		71
1	32		72
6	33		73
=	34		74
stop	35		75
goto	36		76
1	37		77
0	38		78
	39		79

MODULAR ARITHMETIC

KEY	#	KEY	#
HALT	00		40
+	01		41
rcl	02		42
0	03		43
=	04		44
gin	05		45
0	06		46
4	07		47
+/−	08		48
+	09		49
rcl	10		50
0	11		51
=	12		52
gin	13		53
1	14		54
2	15		55
+/−	16		56
gin	17		57
2	18		58
3	19		59
goto	20		60
0	21		61
0	22		62
+	23		63
rcl	24		64
0	25		65
=	26		66
goto	27		67
0	28		68
0	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

This program finds the remainder when one number is divided by another, the modulus.

$a_{\text{mod b}}$ = the remainder when a is divided by b.

The program completes arithmetic operations, so/run/ may be used in the same way as /=/, in order to give addition and subtraction and multiplication mod b.

($+_b$, $-_b$, $\times_b$, respectively)

Pre-execution:

Store b in memory 0.

Execution:

(i) a/run/$a_{\text{mod b}}$

(ii) x + y/run/$x +_b y$

(iii) x × y/run/$x \times_b y$

(iv) x − y/run/$x -_b y$

Warning: long execution times can be expected with large numbers.

MODULAR ARITHMETIC

This program functions in the same way as the previous one. It may, however, return answers such as 4.0000001 or 7.9999999 when the modulus has prime factors other than 2 and 5. It is faster than the previous program.

Store modulus, b, in memory 0.

Execution:

a/run/$a_{\text{mod } b}$
$a_1 \times a_2$/run/$a_1 \times_b a_2$
$a_1 + a_2$/run/$a_1 +_b a_2$
etc.

KEY	#	KEY	#
HALT	00		40
÷	01		41
rcl	02		42
0	03		43
—	04		44
(	05		45
+	06		46
1	07		47
./EE	08		48
./EE	09		49
9	10		50
—	11		51
1	12		52
./EE	13		53
./EE	14		54
9	15		55
)	16		56
×	17		57
rcl	18		58
0	19		59
=	20		60
goto	21		61
0	22		62
0	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

DIAGONAL CODING FUNCTION

KEY	#	KEY	#
HALT	00	x◀▶y	40
+	01	sto	41
(	02	1	42
+	03	1	43
stop	04	M+	44
×	05	0	45
(	06	goto	46
+	07	2	47
1	08	4	48
)	09	rcl	49
÷	10	2	50
2	11	—	51
)	12	rcl	52
=	13	1	53
goto	14	=	54
0	15	stop	55
0	16	+/—	56
sto	17	+	57
2	18	rcl	58
0	19	0	59
sto	20	=	60
0	21	stop	61
sto	22	goto	62
1	23	1	63
rcl	24	7	64
2	25		65
—	26		66
(	27		67
rcl	28		68
0	29		69
+	30		70
rcl	31		71
1	32		72
+	33		73
1	34		74
)	35		75
=	36		76
gin	37		77
4	38		78
9	39		79

Codes pairs of integers as a single integer.

$c = \frac{1}{2}(x + y + 1)(x + y) + x$

Execution:

(a) goto/0/1/x/run/
y/run/c

(b) goto/1/7/c/run/
x/run/y

It is not necessary to repeat the goto/0/1 or goto/1/7 in order to use (a) or (b) again.

POWERS OF 2 CODING FUNCTION

Codes pair (x, y) of integers as

$c = 2^x(2y + 1)$

Execution:

(a) goto/0/1/x/run/y/run/c
(b) goto/1/16/c/run/x/run/y

It is not necessary to repeat the goto/0/1 or the goto/1/6 in order to use (a) or (b) again.

Warning:

0 is a code of no pair under this function.

KEY	#	KEY	#
HALT	00	1	40
y^x	01	./EE	41
2	02	./EE	42
x◀▶y	03	9	43
×	04	)	44
(	05	=	45
stop	06	+/−	46
×	07	gin	47
2	08	5	48
+	09	3	49
1	10	goto	50
)	11	2	51
=	12	2	52
goto	13	rcl	53
0	14	0	54
0	15	stop	55
sto	16	rcl	56
2	17	2	57
1	18	−	58
+/−	19	./EE	59
sto	20	5	60
0	21	=	61
1	22	stop	62
M+	23	goto	63
0	24	1	64
rcl	25	6	65
2	26		66
÷	27		67
2	28		68
=	29		69
sto	30		70
2	31		71
−	32		72
(	33		73
+	34		74
1	35		75
./EE	36		76
./EE	37		77
9	38		78
−	39		79

24. POLYNOMIALS

To solve polynomials use one of the programs from the section on equation solving.

EVALUATION OF A POLYNOMIAL

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

Pre-execution:

goto/0/1

Execution:

x/run/a_n/run/a_{n-1}/
run/a_{n-2}/. . . /a_2/run/
a_1/run/a_0/ = /p(x)

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
×	04		44
rcl	05		45
0	06		46
+	07		47
goto	08		48
0	09		49
3	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

EVALUATION OF A FIXED POLYNOMIAL

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	6	41
0	02	+	42
stop	03	rcl	43
sto	04	1	44
1	05	×	45
stop	06	rcl	46
sto	07	6	47
2	08	+	48
stop	09	rcl	49
sto	10	0	50
3	11	=	51
stop	12	goto	52
sto	13	1	53
4	14	8	54
stop	15		55
sto	16		56
5	17		57
stop	18		58
sto	19		59
6	20		60
×	21		61
rcl	22		62
5	23		63
+	24		64
rcl	25		65
4	26		66
×	27		67
rcl	28		68
6	29		69
+	30		70
rcl	31		71
3	32		72
×	33		73
rcl	34		74
6	35		75
+	36		76
rcl	37		77
2	38		78
×	39		79

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5$$

Execution:

(a) a_0/run/a_1/run/
a_2/run/a_3/run/
a_4/run/a_5/run/

(b) x/run/p(x)/
x'/run/p(x')
etc.

To enter new a_i:
goto/0/1/ and repeat from (a).

To evaluate for new x just repeat from (b).

EVALUATION OF LONG FIXED POLYNOMIAL

$$p(x) = \sum_{i=0}^{n} a_i x^i$$

Store the last six coefficients in memories 0–5.

Enter the remainder of the coefficients as program steps. Use the method of program (2) – i.e. write the polynomial as

$((\ldots((a_n x + a_{n-1})x + a_{n-2})x$

$+ a_{n-3})x + \ldots + a_1)x + a_0$.

Example:

to evaluate

$840x^8 + 516x^7 + 4.7x^6 + 15.4x^5$

$+ 11.81x^4 + 9.2x^3 + 19x^2$

$+ 87x + 1$

Store 840, 516, 4.7, 15.4, 11.81 and 9.2 in memories 0 to 5 respectively and use the program opposite.

Execution:

x/run/p(x)

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	6	41
6	02	+	42
×	03	8	43
rcl	04	7	44
0	05	×	45
+	06	rcl	46
rcl	07	6	47
1	08	+	48
×	09	1	49
rcl	10	=	50
6	11	goto	51
+	12	0	52
rcl	13	0	53
2	14		54
×	15		55
rcl	16		56
6	17		57
+	18		58
rcl	19		59
3	20		60
×	21		61
rcl	22		62
6	23		63
+	24		64
rcl	25		65
4	26		66
×	27		67
rcl	28		68
6	29		69
+	30		70
rcl	31		71
5	32		72
×	33		73
rcl	34		74
6	35		75
+	36		76
1	37		77
9	38		78
×	39		79

DIVISION OF POLYNOMIAL BY LINEAR FACTOR

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
0	03		43
sto	04		44
1	05		45
stop	06		46
+	07		47
rcl	08		48
1	09		49
×	10		50
rcl	11		51
0	12		52
=	13		53
sto	14		54
1	15		55
÷	16		56
rcl	17		57
0	18		58
=	19		59
goto	20		60
0	21		61
6	22		62
+	23		63
rcl	24		64
1	25		65
=	26		66
goto	27		67
0	28		68
0	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$p(x) =$

$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0$

$= (b_{n-1} x^{n-1} + \ldots + b_0)(x - t) + r.$

This program gives the coefficients b_i of $\frac{p(x)}{x-t}$ and the remainder r (which equals p(t)).

Execution:

t/run/

a_n/run/ b_{n-1}

a_{n-1}/run/ b_{n-2}

a_{n-2}/run/ b_{n-3}

⋮

a_2/run/ b_1

a_1/run/ b_0

goto/2/3/a_0/run/ r

DIVISION OF A POLYNOMIAL BY A QUADRATIC

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

$$q(x) = x^2 + c_1 x + c_0$$

$$\frac{p(x)}{q(x)} = t(x) + r(x)$$

$$t(x) = b_{n-2} x^{n-2} + b_{n-3} x^{n-3} + \ldots + b_1 x + b_0$$

$$r(x) = r_1 x + r_0$$

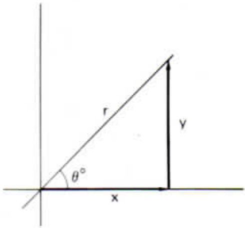
Execution:

c_1/run/c_0/
run/a_n/run/b_{n-2}
a_{n-1}/run/b_{n-3}
a_{n-2}/run/b_{n-4}
.... a_2/run/b_0
a_1/run/r_1
goto/4/0/a_0/
run/r_0

KEY	#	KEY	#
HALT	00	−	40
sto	01	(	41
3	02	rcl	42
stop	03	2	43
sto	04	×	44
4	05	rcl	45
0	06	4	46
sto	07	)	47
1	08	=	48
sto	09	goto	49
2	10	0	50
stop	11	0	51
−	12		52
(	13		53
rcl	14		54
1	15		55
×	16		56
rcl	17		57
3	18		58
)	19		59
−	20		60
(	21		61
rcl	22		62
2	23		63
×	24		64
rcl	25		65
4	26		66
)	27		67
=	28		68
x◀▶y	29		69
rcl	30		70
1	31		71
sto	32		72
2	33		73
x◀▶y	34		74
sto	35		75
1	36		76
goto	37		77
1	38		78
1	39		79

25. CO-ORDINATE GEOMETRY

POLAR / CARTESIAN CO-ORDINATE CONVERSION



Cartesian to polar:

Pre-execution:

goto/0/1

Execution:

x/run/y/run/r/run/θ°

It is not necessary to repeat the pre-execution sequences for subsequent executions of the same sort of conversion.

Polar to cartesian:

Pre-execution:

goto/5/1

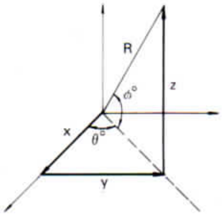
Execution:

r/run/θ/run/x/run/y

KEY	#	KEY	#
HALT	00	goto	40
sto	01	0	41
0	02	0	42
stop	03		43
sto	04		44
1	05		45
x^2	06		46
+	07		47
rcl	08		48
0	09		49
x^2	10	stop	50
=	11	×	51
$\sqrt{x}$	12	stop	52
stop	13	sto	53
1/x	14	0	54
×	15	cos	55
rcl	16	x ◀▶ y	56
0	17	=	57
=	18	stop	58
arc	19	rcl	59
cos	20	0	60
sto	21	sin	61
0	22	=	62
rcl	23	goto	63
1	24	5	64
gin	25	0	65
3	26		66
3	27		67
rcl	28		68
0	29		69
goto	30		70
0	31		71
0	32		72
3	33		73
6	34		74
0	35		75
—	36		76
rcl	37		77
0	38		78
=	39		79

CARTESIAN TO SPHERICAL POLAR CO-ORDINATES

KEY	#	KEY	#
HALT	00	stop	40
sto	01	rcl	41
0	02	3	42
x^2	03	÷	43
+	04	rcl	44
stop	05	5	45
sto	06	=	46
1	07	arc	47
x^2	08	sin	48
=	09	goto	49
sto	10	0	50
2	11	0	51
+	12	3	52
stop	13	6	53
sto	14	0	54
3	15	−	55
x^2	16	rcl	56
=	17	0	57
$\sqrt{x}$	18	=	58
sto	19	goto	59
5	20	4	60
stop	21	0	61
rcl	22		62
0	23		63
÷	24		64
rcl	25		65
2	26		66
$\sqrt{x}$	27		67
=	28		68
arc	29		69
cos	30		70
sto	31		71
0	32		72
rcl	33		73
1	34		74
gin	35		75
5	36		76
2	37		77
rcl	38		78
0	39		79



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

$$\theta = \text{arc tan}\,\frac{y}{x}$$

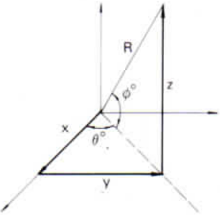
$$\phi = \text{arc sin}\,\frac{z}{R}$$

The value of θ is given in the range 0° to 360°, the value of ϕ in the range −90° to 90°.

Execution:

x/run/y/run/z/run/
R/run/θ/run/ϕ

SPHERICAL POLAR TO CARTESIAN CO-ORDINATES



$x = R \cos\theta \cos\phi$
$y = R \sin\theta \cos\phi$
$z = R \sin\phi$

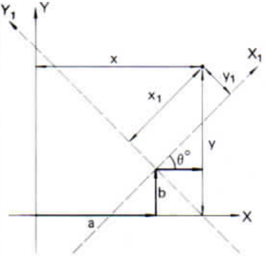
Execution:

R/run/$\theta°$/run/$\phi°$/run/x/run/y/run/z

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
cos	09		49
×	10		50
rcl	11		51
0	12		52
×	13		53
rcl	14		54
1	15		55
cos	16		56
x◀▶y	17		57
=	18		58
stop	19		59
rcl	20		60
1	21		61
sin	22		62
=	23		63
stop	24		64
rcl	25		65
0	26		66
×	27		67
rcl	28		68
2	29		69
sin	30		70
=	31		71
goto	32		72
0	33		73
0	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

TRANSLATION AND ROTATION OF CO-ORDINATES

KEY	#	KEY	#
HALT	00	—	40
sto	01	rcl	41
3	02	0	42
stop	03	×	43
sto	04	rcl	44
4	05	2	45
—	06	sin	46
rcl	07	)	47
1	08	=	48
×	09	stop	49
rcl	10	goto	50
2	11	0	51
sin	12	1	52
+	13		53
(	14		54
rcl	15		55
3	16		56
—	17		57
rcl	18		58
0	19		59
×	20		60
rcl	21		61
2	22		62
cos	23		63
)	24		64
=	25		65
stop	26		66
rcl	27		67
4	28		68
—	29		69
rcl	30		70
1	31		71
×	32		72
rcl	33		73
2	34		74
cos	35		75
—	36		76
(	37		77
rcl	38		78
3	39		79



Across by a, up by b, round by $\theta°$

Pre-execution:

a/sto/0
b/sto/1
θ/sto/2

Execution:

x/run/y/run/x_1/run/y_1

Example:

a = 1
b = 1
$\theta = 45^\circ$

Store in appropriate memories.

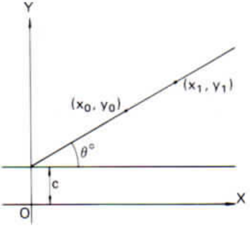
1/run/1/run/0/run/0
4/run/5/run/4.9497474/run/
7.0710–01
3/run/2/run/2.1213204/run/
– 7.0710–01
etc.

So under this transformation:

(1, 1) → (0, 0)
(4, 5) → (4.9497, .7071)
(3, 2) → (2.121, – .7071)
etc.

INTERCEPT AND SLOPE OF A LINE GIVEN TWO POINTS ON IT

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
−	10		50
rcl	11		51
1	12		52
÷	13		53
(	14		54
rcl	15		55
2	16		56
−	17		57
rcl	18		58
0	19		59
)	20		60
=	21		61
sto	22		62
2	23		63
arc	24		64
tan	25		65
stop	26		66
rcl	27		67
2	28		68
×	29		69
rcl	30		70
0	31		71
+/−	32		72
+	33		73
rcl	34		74
1	35		75
=	36		76
goto	37		77
0	38		78
0	39		79



θ is given in the range $(-90°, 90°)$.

If Error is returned then slope = 90° and there is no intercept: C/goto/0/1 before re-using.

Execution:

x_0/run/y_0/run/x_1/run/y_1/run/$\theta°$/run/

c.

AREA OF TRIANGLE FROM CO-ORDINATES OF VERTICES

$$A = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \end{vmatrix}$$

Execution:

x_0/run/y_0/run/x_1/run/y_1/run/x_2/run/y_2/run/A

KEY	#	KEY	#
HALT	00	×	40
sto	01	rcl	41
0	02	1	42
stop	03	)	43
sto	04	÷	44
1	05	2	45
stop	06	=	46
sto	07	x^2	47
2	08	$\sqrt{x}$	48
stop	09	goto	49
sto	10	0	50
3	11	0	51
+/−	12		52
+	13		53
rcl	14		54
1	15		55
×	16		56
stop	17		57
+	18		58
(	19		59
rcl	20		60
2	21		61
−	22		62
rcl	23		63
0	24		64
×	25		65
stop	26		66
)	27		67
+	28		68
(	29		69
rcl	30		70
0	31		71
×	32		72
rcl	33		73
3	34		74
)	35		75
−	36		76
(	37		77
rcl	38		78
2	39		79

KEY	#	KEY	#
HALT	00		40
x^2	01		41
+	02		42
1	03		43
y^x	04		44
1	05		45
.	06		46
5	07		47
÷	08		48
stop	09		49
=	10		50
goto	11		51
0	12		52
0	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

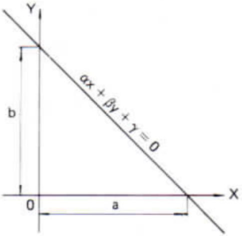
RADIUS OF CURVATURE

$$\rho = \frac{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{3/2}}{\frac{d^2y}{dx^2}}$$

Execution:

$\frac{dy}{dx}$ / run / $\frac{d^2y}{dx^2}$ / run / ρ

INTERCEPTS OF A LINE



$$\alpha x + \beta y + \gamma = 0$$

$$a = -\frac{\gamma}{\alpha}$$

$$b = -\frac{\gamma}{\beta}$$

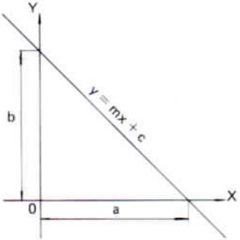
Execution:

α/run/β/run/γ/
run/b/run/a

KEY	#	KEY	#
HALT	00		40
+/−	01		41
sto	02		42
0	03		43
stop	04		44
+/−	05		45
÷	06		46
stop	07		47
=	08		48
1/x	09		49
stop	10		50
rcl	11		51
0	12		52
=	13		53
1/x	14		54
goto	15		55
0	16		56
0	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

EQUATION OF LINE FROM INTERCEPTS

KEY	#	KEY	#
HALT	00		40
1/x	01		41
×	02		42
stop	03		43
+/−	04		44
=	05		45
goto	06		46
0	07		47
0	08		48
	09		49
	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79



$$m = -\frac{b}{a}, c = b$$

Execution:

a/run/b/run/m

Error will be returned if a = 0 (infinite slope).

DISTANCE BETWEEN POINTS IN 3-SPACE

Distance between (x_1, y_1, z_1) and (x_2, y_2, z_2),
$d =$
$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$

Execution:

(a) x_1/run/y_1/run/z_1
run/x_2/run/y_2/
run/z_2/run/ d

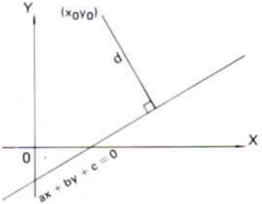
(b) to repeat with same (x_1, y_1, z_1):
x_2/run/y_2/run/
z_2/run/ d

(c) to repeat with new (x_1, y_1, z_1):
goto/0/1
then as in (a).

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
−	10		50
rcl	11		51
0	12		52
=	13		53
x^2	14		54
+	15		55
(	16		56
stop	17		57
−	18		58
rcl	19		59
1	20		60
)	21		61
x^2	22		62
+	23		63
(	24		64
stop	25		65
−	26		66
rcl	27		67
2	28		68
)	29		69
x^2	30		70
=	31		71
$\sqrt{x}$	32		72
goto	33		73
0	34		74
9	35		75
	36		76
	37		77
	38		78
	39		79

DISTANCE OF A POINT FROM A LINE

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
×	10		50
rcl	11		51
0	12		52
+	13		53
(	14		54
stop	15		55
×	16		56
rcl	17		57
1	18		58
)	19		59
+	20		60
rcl	21		61
2	22		62
÷	23		63
(	24		64
rcl	25		65
0	26		66
x^2	27		67
+	28		68
rcl	29		69
1	30		70
x^2	31		71
)	32		72
$\sqrt{x}$	33		73
=	34		74
x^2	35		75
$\sqrt{x}$	36		76
goto	37		77
0	38		78
9	39		79



$$d = \left| \frac{ax_0 + by_0 + c}{\sqrt{a^2 + b^2}} \right|$$

Execution:

(a) a/run/b/run/c /run/x_0/run/y_0/ run/d

(b) to repeat with same a, b, c: new x_0/run/new y_0/run/ new d

(c) to repeat with new a, b, c: goto/0/1 then as in (a)

26. VECTORS, MATRICES AND DETERMINANTS

26.1 Produce of 2×2 – matrices

26.2 Left product of 2×2 – matrix with fixed 2×2 – matrix

26.3 Right product of 2×2 – matrix with fixed 2×2 matrix

26.4 Product of 2×2 – matrix and 2 – vector

26.5 Product of 2 – vector by fixed 2×2 – matrix

26.6 Determinant of 2×2 matrix

26.7 Inverse of 2×2 matrix

26.8 Dot product of 3 – vectors

26.9 Dot product 3 – vector with fixed 3 – vector

26.10 Crossproduct of two 3 – vectors

26.11 Tensor product of two 3 – vectors

26.12 Multiply 3×3 matrices

26.13 Product of 3×3 matrix with a 3 vector

26.14 Determinant of a 3×3 matrix

26.15 Inverting a 3×3 matrix

26.16 Vector Triple Product

PRODUCT OF 2 x 2 - MATRICES

$$\begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}\begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} = \begin{pmatrix} a_3 & b_3 \\ c_3 & d_3 \end{pmatrix}$$

Execution:

a_1/run/b_1/
run/c_1/run/d_1/
run/a_2/run/b_2/
run/c_2/run/d_2/
run/a_3/run/b_3/
run/c_3/run/d_3

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	3	41
0	02	=	42
stop	03	M+	43
sto	04	5	44
1	05	rcl	45
stop	06	3	46
sto	07	×	47
2	08	stop	48
stop	09	=	49
sto	10	M+	50
3	11	2	51
rcl	12	rcl	52
2	13	1	53
×	14	=	54
stop	15	M+	55
=	16	0	56
sto	17	rcl	57
5	18	1	58
rcl	19	)	59
0	20	)	60
=	21	=	61
+	22	stop	62
(	23	rcl	63
rcl	24	0	64
0	25	stop	65
×	26	rcl	66
stop	27	5	67
=	28	stop	68
sto	29	rcl	69
0	30	2	70
rcl	31	goto	71
2	32	0	72
=	33	0	73
sto	34		74
2	35		75
stop	36		76
×	37		77
(	38		78
×	39		79

PRODUCT OF 2x2-MATRIX WITH FIXED 2x2-MATRIX (ON THE LEFT)

KEY	#	KEY	#
HALT	00	4	40
×	01	rcl	41
rcl	02	1	42
2	03	=	43
x◀▶y	04	M+	44
=	05	6	45
sto	06	rcl	46
5	07	1	47
rcl	08	)	48
0	09	)	49
=	10	=	50
+	11	stop	51
(	12	rcl	52
rcl	13	6	53
0	14	stop	54
×	15	rcl	55
stop	16	5	56
=	17	stop	57
sto	18	rcl	58
6	19	4	59
rcl	20	goto	60
2	21	0	61
=	22	0	62
sto	23		63
4	24		64
stop	25		65
×	26		66
(	27		67
×	28		68
rcl	29		69
3	30		70
=	31		71
M+	32		72
5	33		73
rcl	34		74
3	35		75
×	36		76
stop	37		77
=	38		78
M+	39		79

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix} = \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix}$$

Store a, b, c, d in memories 0, 1, 2, 3 respectively.

Execution:

a_1/run/b_1/
run/c_1/run/d_1/
run/a_2/run/b_2/
run/c_2/run/d_2

Program may be re-run with different a_1, b_1, c_1, d_1 without re-entering a, b, c, d.

PRODUCT OF 2x2-MATRIX WITH FIXED 2x2-MATRIX (ON THE RIGHT)

$$\begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix}$$

Store a, b, c, d in memories 0, 1, 2, 3 respectively.

Execution:

a_1/run/b_1/
run/c_1/run/d_1/
run/a_2/run/b_2/
run/c_2/run/d_2

Program may be re-run with different a_1, b_1, c_1, d_1 without re-entering a, b, c, d.

KEY	#	KEY	#
HALT	00	stop	40
×	01	=	41
rcl	02	M+	42
0	03	6	43
x◀▶y	04	rcl	44
=	05	3	45
sto	06	=	46
4	07	)	47
rcl	08	=	48
1	09	x◀▶y	49
=	10	rcl	50
sto	11	4	51
5	12	x◀▶y	52
rcl	13	sto	53
2	14	4	54
×	15	x◀▶y	55
stop	16	stop	56
=	17	rcl	57
M+	18	5	58
4	19	stop	59
rcl	20	rcl	60
3	21	6	61
=	22	stop	62
M+	23	rcl	63
5	24	4	64
rcl	25	goto	65
0	26	0	66
×	27	0	67
stop	28		68
=	29		69
sto	30		70
6	31		71
rcl	32		72
1	33		73
=	34		74
+	35		75
(	36		76
rcl	37		77
2	38		78
×	39		79

PRODUCT OF 2x2-MATRIX AND 2-VECTOR

KEY	#	KEY	#
HALT	00	5	40
sto	01	goto	41
0	02	0	42
stop	03	0	43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
sto	10		50
3	11		51
rcl	12		52
0	13		53
×	14		54
stop	15		55
=	16		56
sto	17		57
4	18		58
rcl	19		59
2	20		60
=	21		61
sto	22		62
5	23		63
rcl	24		64
1	25		65
×	26		66
stop	27		67
=	28		68
M+	29		69
4	30		70
rcl	31		71
3	32		72
=	33		73
M+	34		74
5	35		75
rcl	36		76
4	37		77
stop	38		78
rcl	39		79

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Execution:

a/run/b/run/c/run/d/
run/x/run/y/
run/x'/run/y'

MULTIPLICATION OF 2-VECTOR BY FIXED 2x2-MATRIX

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Store a, b, c, d in memories 0, 1, 2, 3 respectively.

Execution:

x/run/y/run/
x'/run/y'

It is not necessary to re-enter a, b, c, d to re-run with different x, y.

KEY	#	KEY	#
HALT	00		40
×	01		41
rcl	02		42
0	03		43
x◀▶y	04		44
=	05		45
sto	06		46
4	07		47
rcl	08		48
2	09		49
=	10		50
sto	11		51
5	12		52
rcl	13		53
1	14		54
×	15		55
stop	16		56
=	17		57
M+	18		58
4	19		59
rcl	20		60
3	21		61
=	22		62
M+	23		63
5	24		64
rcl	25		65
4	26		66
stop	27		67
rcl	28		68
5	29		69
goto	30		70
0	31		71
0	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

DETERMINANT OF A 2x2-MATRIX

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
×	04		44
stop	05		45
+/−	06		46
+	07		47
(	08		48
stop	09		49
×	10		50
rcl	11		51
0	12		52
)	13		53
=	14		54
goto	15		55
0	16		56
0	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$$\Delta = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

Execution:

a/run/b/run/c/
run/d/run/ Δ

INVERSE OF 2x2-MATRIX

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix}$$

Execution:

a/run/b/run/c/run/d/run/a'/run/b'/run/c'/run/d'

Gives error if determinant = 0.

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	3	41
1	02	+/−	42
stop	03	=	43
sto	04	stop	44
2	05	rcl	45
stop	06	1	46
sto	07	=	47
3	08	goto	48
stop	09	0	49
sto	10	0	50
4	11		51
rcl	12		52
1	13		53
×	14		54
rcl	15		55
4	16		56
−	17		57
(	18		58
rcl	19		59
2	20		60
×	21		61
rcl	22		62
3	23		63
)	24		64
=	25		65
sto	26		66
5	27		67
rcl	28		68
4	29		69
÷	30		70
rcl	31		71
5	32		72
=	33		73
stop	34		74
rcl	35		75
2	36		76
+/−	37		77
=	38		78
stop	39		79

DOT PRODUCT OF 3-VECTORS

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
×	10		50
rcl	11		51
0	12		52
+	13		53
(	14		54
stop	15		55
×	16		56
rcl	17		57
1	18		58
)	19		59
+	20		60
(	21		61
stop	22		62
×	23		63
rcl	24		64
2	25		65
)	26		66
=	27		67
goto	28		68
0	29		69
0	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$$(x_1, y_1, z_1)\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}$$

$$= x_1x_2 + y_1y_2 + z_1z_2$$

Execution:

x_1/run/y_1/run/z_1/run/x_2/run/y_2/run/z_2/run/product

DOT PRODUCT OF 3-VECTOR WITH FIXED 3-VECTOR

$$(a, b, c)\begin{pmatrix} x \\ y \\ z \end{pmatrix} = ax + by + cz$$

Store a, b, c in memories 0, 1, 2 respectively.

Execution:

x/run/y/run/z/
run/product.

The program may be re-run for different x, y, z without re-entering a, b, c.

KEY	#	KEY	#
HALT	00		40
×	01		41
rcl	02		42
0	03		43
+	04		44
(	05		45
stop	06		46
×	07		47
rcl	08		48
1	09		49
)	10		50
+	11		51
(	12		52
stop	13		53
×	14		54
rcl	15		55
2	16		56
)	17		57
=	18		58
goto	19		59
0	20		60
0	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

CROSS PRODUCT OF TWO 3-VECTORS

KEY	#	KEY	#
HALT	00	×	40
sto	01	rcl	41
0	02	5	42
stop	03	)	43
sto	04	=	44
1	05	stop	45
stop	06	rcl	46
sto	07	0	47
2	08	×	48
stop	09	rcl	49
sto	10	4	50
3	11	—	51
stop	12	(	52
sto	13	rcl	53
4	14	3	54
stop	15	×	55
sto	16	rcl	56
5	17	1	57
×	18	)	58
rcl	19	=	59
1	20	goto	60
—	21	0	61
(	22	0	62
rcl	23		63
4	24		64
×	25		65
rcl	26		66
2	27		67
)	28		68
=	29		69
stop	30		70
rcl	31		71
3	32		72
×	33		73
rcl	34		74
2	35		75
—	36		76
(	37		77
rcl	38		78
0	39		79

$(x, y, z) \wedge (x', y', z')$
$= (yz' - y'z,\ x'z - xz',\ xy' - x'y)$
$= (p_1, p_2, p_3)$

Execution:

x/run/y/run/z/run/
x'/run/y'/run/z'/
run/p_1/run/p_2/
run/p_3

TENSOR PRODUCT OF TWO 3-VECTORS

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} (y_1, y_2, y_3) = \begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ x_2y_1 & x_2y_2 & x_2y_3 \\ x_3y_1 & x_3y_2 & x_3y_3 \end{pmatrix}$$

Pre-execution:

goto/0/1

Execution:

y_1/run/y_2/run/y_3/run/x_1/run/x_1y_1 run/x_1y_2/run/x_1y_3 x_2/run/x_2y_1/run/x_2y_2/run/x_2y_3/ x_3/run/x_3y_1/run /x_3y_2/run/x_3y_3

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
×	10		50
rcl	11		51
0	12		52
x◀▶y	13		53
=	14		54
stop	15		55
rcl	16		56
1	17		57
=	18		58
stop	19		59
rcl	20		60
2	21		61
=	22		62
goto	23		63
0	24		64
9	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

MULTIPLYING 3 x 3 MATRICES

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
×	10		50
rcl	11		51
0	12		52
+	13		53
(	14		54
stop	15		55
×	16		56
rcl	17		57
1	18		58
)	19		59
+	20		60
(	21		61
stop	22		62
×	23		63
rcl	24		64
2	25		65
)	26		66
=	27		67
goto	28		68
0	29		69
9	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

This program finds row i of C given B and row i of A.

Pre-execution:

goto/0/1

Execution:

a_{i1}/run/a_{i2}/run/
a_{i3}/run/b_{11}/run/
b_{21}/run/b_{31}/run/
c_{i1}/b_{12}/run/b_{22}/
run/b_{32}/run/c_{i2}/
b_{13}/run/b_{23}/run/
b_{33}/run/c_{i3}

PRODUCT OF A 3x3 MATRIX WITH A 3-VECTOR

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

Pre-execution:

goto/0/1

Execution:

x/run/y/run/z/
run/a_{11}/run/a_{12}
run/a_{13}/run/x'/
a_{21}/run/a_{22}/run/
a_{23}/run/y'/a_{31}
/run/a_{32}/run/
a_{33}/z'

KEY	#	KEY	#
HALT	00		40
sto	01		41
0	02		42
stop	03		43
sto	04		44
1	05		45
stop	06		46
sto	07		47
2	08		48
stop	09		49
×	10		50
rcl	11		51
0	12		52
+	13		53
(	14		54
stop	15		55
×	16		56
rcl	17		57
1	18		58
)	19		59
+	20		60
(	21		61
stop	22		62
×	23		63
rcl	24		64
2	25		65
)	26		66
=	27		67
goto	28		68
0	29		69
9	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

DETERMINANT OF A 3x3 MATRIX

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	0	41
0	02	×	42
stop	03	rcl	43
sto	04	5	44
1	05	)	45
stop	06	×	46
sto	07	stop	47
2	08	)	48
stop	09	+	49
sto	10	(	50
3	11	rcl	51
stop	12	0	52
sto	13	×	53
4	14	rcl	54
stop	15	4	55
sto	16	—	56
5	17	(	57
×	18	rcl	58
rcl	19	1	59
1	20	×	60
—	21	rcl	61
(	22	3	62
rcl	23	)	63
2	24	×	64
×	25	stop	65
rcl	26	)	66
4	27	=	67
)	28	goto	68
×	29	0	69
stop	30	0	70
+	31		71
(	32		72
rcl	33		73
2	34		74
×	35		75
rcl	36		76
3	37		77
—	38		78
(	39		79

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{33} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Execution:

a_{11}/run/a_{12}/run/a_{13}/run
a_{21}/run/a_{22}/run/a_{23}/run/
a_{31}/run/a_{32}/run/a_{33}
/run/Δ

INVERTING A 3x3-MATRIX

This program finds the adjoint matrix and the determinant of a matrix

$A = (a_{ij})$.

Then

$A^{-1} = \frac{1}{|A|} \text{adj } A$.

$\text{adj}A = (\hat{a}_{ij})$.

Execution:

a_{21}/run/a_{22}/run/a_{23}/run/
a_{31}/run/a_{32}/run/a_{33}/run/
$\hat{a}_{11}$/run/$\hat{a}_{21}$/run/$\hat{a}_{32}$/
a_{31}/run/a_{32}/run/a_{33}/run/
a_{11}/run/a_{12}/run/a_{13}/run/
$\hat{a}_{12}$/run/$\hat{a}_{22}$/run/$\hat{a}_{32}$/
a_{11}/run/a_{12}/run/a_{13}/run/
a_{21}/run/a_{22}/run/a_{23}/run/
$\hat{a}_{31}$/run/$\hat{a}_{32}$/run/$\hat{a}_{33}$

goto/6/5/a_{11}/run/$\hat{a}_{11}$/run/a_{12}/
run/a_{21}/run/a_{13}/run/$\hat{a}_{31}$/run/$|A|$

Now

$A^{-1} = \frac{1}{|A|} \text{adj } A$

$A^{-1} = (a^{-1}_{ij})$

So

$\hat{a}_{11}$/÷/$|A|$/=/a^{-1}_{11}/$\hat{a}_{12}$/=/a^{-1}_{12}/
$\hat{a}_{13}$/=/a^{-1}_{13} . . . etc.

KEY	#	KEY	#
HALT	00	×	40
sto	01	rcl	41
0	02	5	42
stop	03	)	43
sto	04	=	44
1	05	stop	45
stop	06	rcl	46
sto	07	0	47
2	08	×	48
stop	09	rcl	49
sto	10	4	50
3	11	—	51
stop	12	(	52
sto	13	rcl	53
4	14	1	54
stop	15	×	55
sto	16	rcl	56
5	17	3	57
×	18	)	58
rcl	19	=	59
1	20	goto	60
—	21	0	61
(	22	0	62
rcl	23		63
2	24		64
×	25	×	65
rcl	26	stop	66
4	27	+	67
)	28	(	68
=	29	stop	69
stop	30	×	70
rcl	31	stop	71
2	32	)	72
×	33	+	73
rcl	34	(	74
3	35	stop	75
—	36	×	76
(	37	stop	77
rcl	38	)	78
0	39	=	79

TRIPLE VECTOR PRODUCT

KEY	#	KEY	#
HALT	00	rcl	40
sto	01	0	41
0	02	×	42
stop	03	rcl	43
sto	04	5	44
1	05	)	45
stop	06	=	46
sto	07	stop	47
2	08	sto	48
stop	09	5	49
sto	10	rcl	50
3	11	0	51
stop	12	×	52
sto	13	rcl	53
4	14	4	54
stop	15	—	55
sto	16	(	56
5	17	rcl	57
×	18	1	58
rcl	19	×	59
1	20	rcl	60
—	21	3	61
(	22	)	62
rcl	23	=	63
2	24	stop	64
×	25	sto	65
rcl	26	0	66
4	27	rcl	67
)	28	5	68
=	29	sto	69
stop	30	1	70
sto	31	rcl	71
6	32	6	72
rcl	33	sto	73
2	34	2	74
×	35	goto	75
rcl	36	0	76
3	37	9	77
—	38		78
(	39		79

This program computes the triple vector product
$\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$
of three vectors **a**, **b**, and **c**.

The vectors a, b and c are represented in the form
$\mathbf{a} = a_1 \mathbf{i} + a_2 \mathbf{j} + a_3 \mathbf{k}$
$\mathbf{b} = b_1 \mathbf{i} + b_2 \mathbf{j} + b_3 \mathbf{k}$
$\mathbf{c} = c_1 \mathbf{i} + c_2 \mathbf{j} + c_3 \mathbf{k}$
where **i**, **j** and **k** are, respectively unit vectors in the directions of the x—, y— and z— axes of a rectangular Cartesian system.
The triple vector product is represented by
$\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}$

Execution:

c_1/run/c_2/run/c_3/run/b_1/run/b_2/run/b_3/run/run/run/run/a_1/run/a_2/run/a_3/run/P/run/Q/run/R

Example:

3/run/2/run/1/run/1/run/2/run/3/run/run/run/run/—12/run/0/run/12

27. TRIGONOMETRY

All angles are in degrees

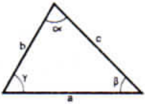
To solve a spherical triangle:

Given	Use
3 sides	Program 5
2 sides and included angle	Program 5
2 sides and other angle	Program 7
2 angles and included side	Program 6
2 angles and other side	Program 7
3 angles	Program 6

Warning:

Ambiguous solutions are sometimes possible, though the programs only return one solution. Remember to make any necessary corrections.

THE SOLUTION OF TRIANGLES



Sine formula:

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}$$

Cosine formula:

$$a^2 = b^2 + c^2 - 2bc\cos\alpha$$

Execution:

(a) goto/0/1/α/run/β/run/
$\frac{a}{b}$/b/run/a

(b) goto/1/2/b/run/
c/run/α/run/a

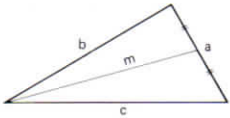
(c) goto/3/8/a/run/
b/run/c/run/α

It is not necessary to repeat the goto to continue with executions of the same type.

KEY	#	KEY	#
HALT	00	+	40
sin	01	stop	41
÷	02	sto	42
stop	03	0	43
sin	04	x^2	44
×	05	+	45
stop	06	stop	46
=	07	sto	47
goto	08	1	48
0	09	x^2	49
0	10	÷	50
stop	11	2	51
sto	12	÷	52
0	13	rcl	53
x^2	14	0	54
+	15	÷	55
stop	16	rcl	56
sto	17	1	57
1	18	=	58
x^2	19	arc	59
−	20	cos	60
(	21	stop	61
2	22	goto	62
×	23	3	63
rcl	24	8	64
0	25		65
×	26		66
rcl	27		67
1	28		68
×	29		69
stop	30		70
cos	31		71
)	32		72
=	33		73
$\sqrt{x}$	34		74
goto	35		75
1	36		76
1	37		77
x^2	38		78
+/−	39		79

MEDIAN OF TRIANGLE

KEY	#	KEY	#
HALT	00		40
x^2	01		41
+/−	02		42
+	03		43
(	04		44
stop	05		45
x^2	06		46
+	07		47
stop	08		48
x^2	09		49
×	10		50
2	11		51
)	12		52
=	13		53
$\sqrt{x}$	14		54
÷	15		55
2	16		56
=	17		57
goto	18		58
0	19		59
0	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

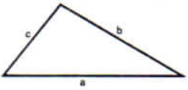


$$m = \frac{\sqrt{2(b^2 + c^2) - a^2}}{2}$$

Execution:

a/run/b/run/c/
run/m

AREA OF TRIANGLE FROM LENGTH OF SIDES



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

$$S = \frac{a+b+c}{2}$$

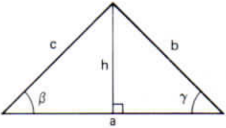
Execution:

a/run/b/run/c/run/A

KEY	#	KEY	#
HALT	00	$\sqrt{x}$	40
sto	01	goto	41
0	02	0	42
+	03	0	43
stop	04		44
sto	05		45
1	06		46
+	07		47
stop	08		48
sto	09		49
2	10		50
=	11		51
÷	12		52
2	13		53
=	14		54
sto	15		55
3	16		56
×	17		57
(	18		58
—	19		59
rcl	20		60
1	21		61
)	22		62
×	23		63
(	24		64
rcl	25		65
3	26		66
—	27		67
rcl	28		68
0	29		69
)	30		70
×	31		71
(	32		72
rcl	33		73
3	34		74
—	35		75
rcl	36		76
2	37		77
)	38		78
=	39		79

HEIGHT AND AREA OF NON-RIGHT ANGLED TRIANGLE

KEY	#	KEY	#
HALT	00		40
×	01		41
stop	02		42
sin	03		43
×	04		44
stop	05		45
÷	06		46
2	07		47
=	08		48
goto	09		49
0	10		50
0	11		51
stop	12		52
×	13		53
(	14		54
÷	15		55
(	16		56
stop	17		57
tan	18		58
1/x	19		59
+	20		60
stop	21		61
tan	22		62
1/x	23		63
)	24		64
=	25		65
stop	26		66
)	27		67
÷	28		68
2	29		69
=	30		70
goto	31		71
1	32		72
2	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79



Height h $= c \sin \beta$

$$= \frac{a \tan \beta \tan \gamma}{\tan \beta + \tan \gamma}$$

Area A $= \frac{1}{2} ah$

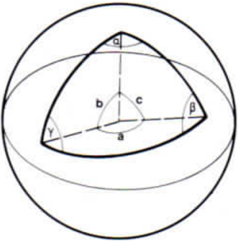
Execution:

(a) goto/0/1/c/run/
β/run/h/a/run/A

(b) goto/1/3/a/run/
β/run/γ/run/h/run/A

It is not necessary to repeat the goto to continue with executions of the same type.

FIRST COSINE FORMULA



$\cos a = \cos b \cos c$
$+ \sin b \sin c \cos \alpha$

Execution:

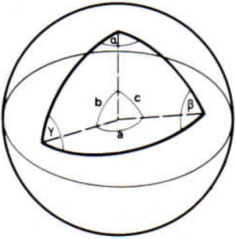
(a) goto/0/1/
α/run/b/run/
c/run/a

(b) goto/4/1/
a/run/b/run/
c/run/α

KEY	#	KEY	#
HALT	00	stop	40
sto	01	sto	41
0	02	0	42
stop	03	stop	43
sto	04	sto	44
1	05	1	45
stop	06	stop	46
sto	07	sto	47
2	08	2	48
sin	09	cos	49
×	10	×	50
rcl	11	rcl	51
1	12	1	52
sin	13	cos	53
×	14	+/−	54
rcl	15	+	55
0	16	rcl	56
cos	17	0	57
+	18	cos	58
(	19	÷	59
rcl	20	rcl	60
1	21	1	61
cos	22	sin	62
×	23	÷	63
rcl	24	rcl	64
2	25	2	65
cos	26	sin	66
)	27	=	67
=	28	arc	68
arc	29	cos	69
cos	30	goto	70
goto	31	4	71
0	32	0	72
0	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

SECOND COSINE FORMULA

KEY	#	KEY	#
HALT	00	stop	40
sto	01	sto	41
0	02	0	42
stop	03	stop	43
sto	04	sto	44
1	05	1	45
stop	06	stop	46
sto	07	sto	47
2	08	2	48
sin	09	cos	49
×	10	×	50
rcl	11	rcl	51
1	12	1	52
sin	13	cos	53
×	14	+	54
rcl	15	rcl	55
0	16	0	56
cos	17	cos	57
−	18	÷	58
(	19	rcl	59
rcl	20	1	60
1	21	sin	61
cos	22	÷	62
×	23	rcl	63
rcl	24	2	64
2	25	sin	65
cos	26	=	66
)	27	arc	67
=	28	cos	68
arc	29	goto	69
cos	30	4	70
goto	31	0	71
0	32		72
0	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79



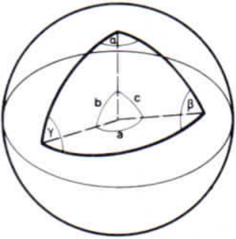
$$\cos\alpha = -\cos\beta\cos\gamma + \sin\beta\sin\gamma\cos a$$

Execution:

(a) goto/0/1/
a/run/β/run/γ/run/α

(b) goto/4/1/
α/run/β/run/γ/run/a

SINE FORMULA AND NAPIER'S FORMULA



$$\frac{\sin a}{\sin \alpha} = \frac{\sin b}{\sin \beta} = \frac{\sin c}{\sin \gamma}$$

$$\tan \frac{a-b}{2} \sin \frac{\alpha+\beta}{2}$$

$$= \sin \frac{\alpha-\beta}{2} \tan \frac{c}{2}$$

Execution:

(a) goto/0/1/a/run/α/run/β/run/b

(b) goto/0/1/α/run/a/run/b/run/β

(c) goto/2/1/a/run/b/run/α/run/β/run/c

KEY	#	KEY	#
HALT	00	)	40
sin	01	÷	41
÷	02	(	42
stop	03	rcl	43
sin	04	0	44
×	05	−	45
stop	06	rcl	46
sin	07	1	47
=	08	÷	48
arc	09	2	49
sin	10	=	50
goto	11	sin	51
0	12	)	52
0	13	=	53
	14	arc	54
	15	tan	55
	16	×	56
	17	2	57
	18	=	58
	19	goto	59
stop	20	2	60
−	21	0	61
stop	22		62
÷	23		63
2	24		64
=	25		65
tan	26		66
×	27		67
(	28		68
stop	29		69
sto	30		70
0	31		71
+	32		72
stop	33		73
sto	34		74
1	35		75
÷	36		76
2	37		77
=	38		78
sin	39		79

AREA OF SPHERICAL TRIANGLE

KEY	#	KEY	#
HALT	00		40
+	01		41
stop	02		42
+	03		43
stop	04		44
=	05		45
D ▶ R	06		46
−	07		47
π	08		48
×	09		49
stop	10		50
x^2	11		51
=	12		52
goto	13		53
0	14		54
0	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

$$A = (\alpha + \beta + \gamma - \pi)R^2$$

Enter α, β, γ in degrees.

N.B. The formula is valid *only* when α, β, γ are in radians. The program takes this into account.

Execution:

α/run/β/run/γ/run/R/run/A

28. CALCULUS

28.1 Numerical integration by the trapezoidal approximation
28.2 Numerical integration by Simpson's Rule
28.3 Numerical integration by Weddle's Formula

For greatest accuracy, Simpson's rule generally gives the best results.

28.4 Integration of function
28.5 Numerical solution of a differential equation – Euler's formula
28.6 Numerical solution of a differential equation – Runge-Kutta formula

No program for numerical differentiation is given as the algorithm chosen depends on the nature of the data to be differentiated. The following, however, should be useful in numerical differentiation:

28.7 Finite differences

TRAPEZOIDAL FORMULA

$$I = \tfrac{1}{2}h(y_0 + 2y_1 + 2y_2 + \ldots + 2y_{n-1} + y_n)$$

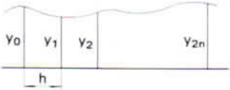
Execution:

y_0/run/y_1/run/. . ./y_{n-1}/run/goto/1/0/y_n/run/h/run/I

KEY	#	KEY	#
HALT	00		40
+	01		41
(	02		42
stop	03		43
×	04		44
2	05		45
)	06		46
goto	07		47
0	08		48
1	09		49
)	10		50
×	11		51
stop	12		52
÷	13		53
2	14		54
=	15		55
goto	16		56
0	17		57
0	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

SIMPSON´S RULE

KEY	#	KEY	#
HALT	00		40
+	01		41
(	02		42
stop	03		43
×	04		44
4	05		45
)	06		46
+	07		47
(	08		48
stop	09		49
×	10		50
2	11		51
)	12		52
+	13		53
goto	14		54
0	15		55
2	16		56
)	17		57
×	18		58
stop	19		59
÷	20		60
3	21		61
=	22		62
goto	23		63
0	24		64
0	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

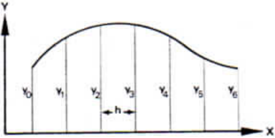


$$I = \frac{1}{3}h\,(y_0 + 4y_1 + 2y_2 + 4y_3 \ldots + 4y_{2n-1} + y_{2n})$$

Execution:

y_0/run/y_1/run/y_2/run/. . ./y_{2n-1}/run/goto/1/7/y_{2n}/run/h/run/ I

WEDDLE FORMULA



$$I = \frac{3h}{10}(y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + y_6)$$

Execution:

y_0/run/y_1/. . . y_5/run/y_6/run/ h/run/I

KEY	#	KEY	#
HALT	00		40
+	01		41
(	02		42
stop	03		43
×	04		44
5	05		45
)	06		46
+	07		47
stop	08		48
+	09		49
(	10		50
stop	11		51
×	12		52
6	13		53
)	14		54
+	15		55
stop	16		56
+	17		57
(	18		58
stop	19		59
×	20		60
5	21		61
)	22		62
+	23		63
stop	24		64
×	25		65
stop	26		66
×	27		67
./EE	28		68
3	29		69
=	30		70
goto	31		71
0	32		72
0	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

INTEGRATION OF A FUNCTION BY TRAPEZOIDAL FORMULA

KEY	#	KEY	#
HALT	00	6	40
sto	01	4	41
0	02	rcl	42
+/−	03	5	43
+	04	+	44
stop	05	rcl	45
÷	06	6	46
9	07	÷	47
9	08	2	48
=	09	×	49
sto	10	rcl	50
2	11	2	51
0	12	=	52
sto	13	M+	53
3	14	3	54
sto	15	9	55
1	16	8	56
Y	17	−	57
O	18	rcl	58
U	19	1	59
R	20	=	60
	21	gin	61
S	22	7	62
E	23	8	63
G	24	rcl	64
M	25	5	65
E	26	sto	66
N	27	6	67
T	28	rcl	68
↓	29	2	69
↓	30	M+	70
↓	31	0	71
sto	32	1	72
5	33	M+	73
rcl	34	1	74
1	35	goto	75
−	36	1	76
1	37	7	77
=	38	rcl	78
gin	39	3	79

$$I = \int_a^b f(x)dx$$

$$h = \frac{b-a}{99}$$

$$I = \frac{h}{2}\{f(a) + 2f(a+h) + \ldots$$
$$+ 2f(a+98h) + f(b)\}$$

Write a program segment to evaluate f(x) where x is in memory 0 – unfortunately only fifteen spaces are available for this.

Execution:

a/run/b/run/I

Execution times of the order of five or ten minutes can be expected.

The number of intervals used in the integration can be changed by changing steps 07, 08, 55 and 56 of the program appropriately.

SOLUTION OF DIFFERENTIAL EQUATIONS

This program determines an approximation to the solution of a differential equation of the form

$$\frac{dy}{dx} = f(x, y)$$

by using Euler's method

$$y(x + h) = y(x) + h\,f(x, y)$$

where h is a step length.

Given an initial condition

$y = a$ when $x = b$

the program successively generates the values of y at the points $x + nh$ for $n = 1, 2 \ldots$

Execution:

h/run/x/run/y/run/$y(x + h)$/run/$y(x + 2h)$/. . .

Example:

This program is written to solve the equation

$$\frac{dy}{dx} = x^2 + y^2$$

Any other equation can be programmed by entering the function f(x, y) from step 27. For the initial conditions $y = 0$ when $x = 0$ the execution sequence with a step length $h = 0.1$ is:

0.1/run/0/run/0/run/0/run/ 1.0×10^{-3}/run/5.0×10^{-3}

KEY	#	KEY	#
HALT	00	0	40
sto	01	=	41
0	02	goto	42
stop	03	1	43
sto	04	2	44
1	05		45
stop	06		46
sto	07		47
2	08		48
goto	09		49
2	10		50
7	11		51
+	12		52
rcl	13		53
2	14		54
=	15		55
stop	16		56
sto	17		57
2	18		58
rcl	19		59
1	20		60
+	21		61
rcl	22		62
0	23		63
=	24		64
sto	25		65
1	26		66
rcl	27		67
1	28		68
x^2	29		69
sto	30		70
3	31		71
rcl	32		72
2	33		73
x^2	34		74
+	35		75
rcl	36		76
3	37		77
×	38		78
rcl	39		79

NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS

KEY	#	KEY	#
HALT	00	5	40
1	01	rcl	41
sto	02	2	42
3	03	M+	43
↑	04	0	44
	05	1	45
	06	+/−	46
	07	goto	47
	08	0	48
Y	09	2	49
O	10		50
U	11		51
R	12		52
	13		53
S	14		54
E	15		55
G	16		56
M	17		57
E	18		58
N	19		59
T	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
↓	27		67
×	28		68
rcl	29		69
2	30		70
÷	31		71
2	32		72
=	33		73
M+	34		74
1	35	rcl	75
rcl	36	0	76
3	37	stop	77
gin	38	rcl	78
7	39	1	79

To solve $\frac{dy}{dx} = f(x, y)$

by the second order Runge-Kutta formula

$$y_{n+1} = y_n + \tfrac{1}{2}hf(x_n, y_n) + \tfrac{1}{2}hf(x_n + h, y_n + \tfrac{1}{2}hf(x_n, y_n))$$

The initial conditions x_0, y_0 must be given, and the step length h, where

$$x_n = x_0 + nh.$$

Write a program segment to evaluate f(x, y) where x is in memory 0, y in memory 1 and insert in the program opposite. E.g. for $f(x, y) = 4x^2 + 5x + 2y + 3xy$ a suitable segment would be

rcl/0/×/4/+/5/×/rcl/0/+/(/rcl/0/×/3/+/2/×/rcl/1/)/.

Your segment can be up to 49 steps long.

Pre-execution:

Store x_0, y_0, h in memories 0, 1, 2 respectively.

Execution:

run/x_1/run/y_1/run/x_2/run/y_2/run/x_3/run/y_3/. . . etc.

FINITE DIFFERENCES

This program finds the forwards differences

$\Delta \quad \Delta^2 \quad \Delta^3$

given data points $x_0\ x_1\ x_2 \ldots$

$\Delta_i = x_{i+1} - x_i$
$\Delta_i^2 = \Delta_{i+1} - \Delta_i$
$\Delta_i^3 = \Delta_{i+1}^2 - \Delta_i^2$

Execution:

x_0/run/run/run/
x_1/run/ Δ_0/run/run/
x_2/run/ Δ_1/run/ Δ_0^2/run/
x_3/run/ Δ_2/run/ Δ_1^2/run/ Δ_0^3/
x_4/run/ Δ_3/run/ Δ_2^2/run/ Δ_1^3/
x_5/run/etc.

Example:

x_i	Δ_i	Δ_i^2	Δ_i^3
5/run	/run	/run	
7/run	/2/run	/run	
9/run	/2/run	/0/run	
8/run/	−1/run/	−3/run/	−3
3/run/	−5/run/	−4/run/	−1
10/run	/7/run	/12/run	/16

etc.

KEY	#	KEY	#
HALT	00		40
−	01		41
rcl	02		42
0	03		43
=	04		44
M+	05		45
0	06		46
stop	07		47
−	08		48
rcl	09		49
1	10		50
=	11		51
M+	12		52
1	13		53
stop	14		54
−	15		55
rcl	16		56
2	17		57
=	18		58
M+	19		59
2	20		60
goto	21		61
0	22		62
0	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

	00		40
	01		41
	02		42
	03		43
	04		44
	05		45
	06		46
	07		47
	08		48
	09		49
	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

	00		40
	01		41
	02		42
	03		43
	04		44
	05		45
	06		46
	07		47
	08		48
	09		49
	10		50
	11		51
	12		52
	13		53
	14		54
	15		55
	16		56
	17		57
	18		58
	19		59
	20		60
	21		61
	22		62
	23		63
	24		64
	25		65
	26		66
	27		67
	28		68
	29		69
	30		70
	31		71
	32		72
	33		73
	34		74
	35		75
	36		76
	37		77
	38		78
	39		79

Sinclair Radionics Inc.
Galleria
115 East 57th Street
New York NY 10022
USA

Sinclair Radionics Ltd.
London Road
St Ives Huntingdon
Cambridgeshire PE17 4HJ

48584 622

sinclair