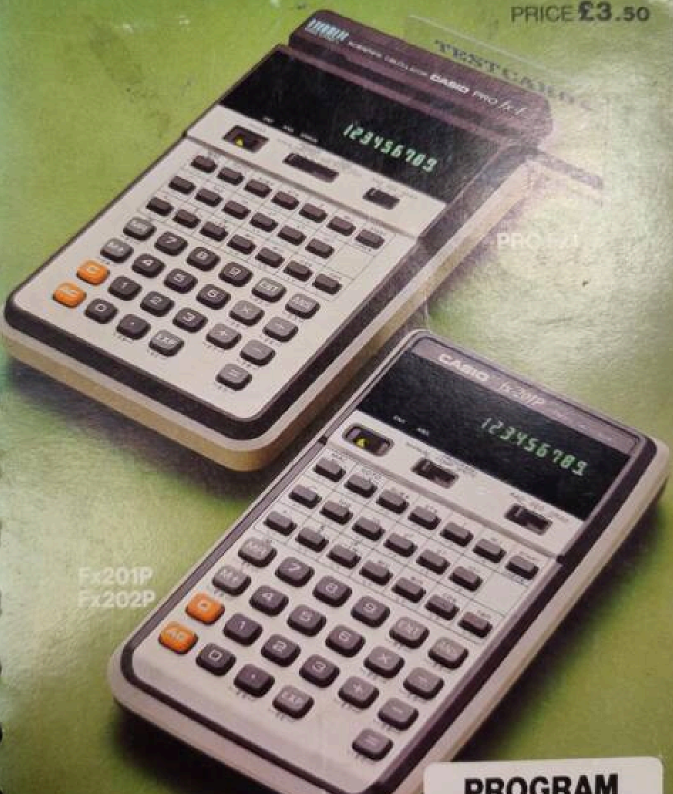
CASIO PROGRAMMABLE SCIENTIFIC CALCULATORS

PRICE £3.50



PROGRAM MANUAL

Casio Program Manual
prepared for Casio Electronics Co. Ltd.
28 Scrutton Street, London EC2A 4TY. Telephone: 01-377 9087.

Written by Robin Bradbeer and Gary Marshall, Department of Electronic and Communications Engineering, The Polytechnic of North London N7.
Production and Artwork prepared by Harrison McCann Advertising Ltd.
Typesetting and make-up by Camera Ready Concepts.
Printed by Colorcraft, London.

CASIO PROGRAMMABLE SCIENTIFIC CALCULATORS

Program Manual

FX-201P. Standard model, 127 step, Fortran method, 10 memory, Scientific, Mains/Battery application.

FX-202P. Deluxe version of FX-201P. All of the same features, plus one year program protection. Separate silver-oxide battery system stores and protects any program for repeated use and against accidental loss.

PRO FX-1. Magnetic card version of FX-201P. All features the same. Programs can be recorded on credit card sized mag. cards.

The logic, features, program length (127 steps), and program method (Fortran) are identical on all three models.

CONTENTS

CONTENTS (continued)

PRO FX-1 MAGNETIC CARD RECORDING INSTRUCTIONS

To Record:

1. Key in program with switch at "WRITE", as on FX-201P/202P.

2. To record program on to a card, set switch to "RECORD".

3. Place mag. card (magnetic strip side facing keyboard) in slot at top of the calculator, drawing it through from left to right.

4. Cover lower strip with anti magnetic tape to prevent accidental erasure and handling damage. Pre-cut anti magnetic strips are packed with all blank cards.

To Read:

1. To load a program from card to calculator, set the switch at "READ".

2. Run the program card through the slot from left to right.

3. To verify that the program has been correctly transferred from card to calculator memory, set the switch to "CHECK", and pass card through a second time. If "ERROR" light stays darkened, the program has been correctly transferred.

4. Set switch at "COMP" and program is ready for use.

Notes:

1. In "RECORD" mode, power consumption is 3 times normal. Weak batteries—still good for calculating—will cause error light to come on, preventing successful recording. Change batteries or use a mains adaptor.

2. Program cards, properly cared for, can be used 1,000 times.

3. After every 100 uses of program cards, pass a cleaning card through the slot.

4. Anti magnetic strips may be removed and cards reused for new programs.

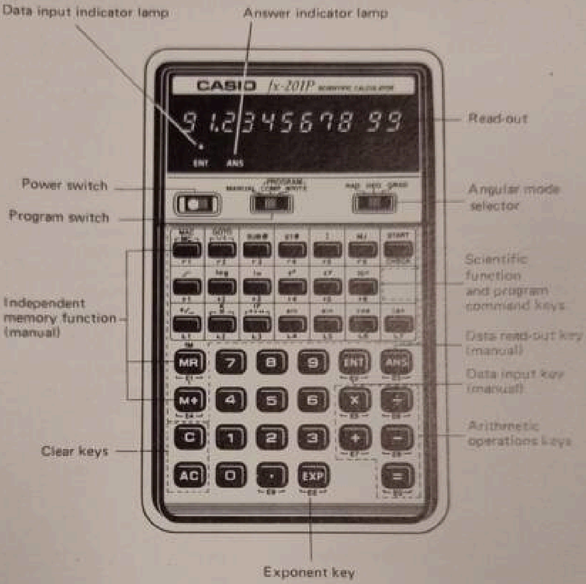
SPECIAL NOTE

IT IS ESSENTIAL THAT BEFORE WRITING A PROGRAM IN THE PRO FX-1 YOU DEPRESS [AC] AND [MAC]

FAILURE TO DO SO WILL PREVENT TRANSFER OF THE PROGRAM TO A MAGNETIC CARD ALTHOUGH IT IS STILL POSSIBLE TO USE THE PROGRAM WITHIN THE MACHINE.

Part 1
MANUAL INSTRUCTIONS

1.1 KEYBOARD – MANUAL FUNCTIONS



The following keys are operational in the manual mode.

+ , − , × , ÷ , =	Arithmetic function keys.
EXP	Enter exponent key.
M+ , MR , MC	Memory function keys.
+/− , C , AC	Change sign and clear keys.
sin , cos , tan , arc	Trigonometric function keys.
log , 10^x , ln , e^x , x^y	Logarithmic function keys.
1/x , π , $\sqrt{\ }$	Reciprocal, pi and square root keys.
°′″	Sexagesimal → decimal conversion key.
ENT , ANS	Data entry and recall keys.

In the manual mode the degrees-radians-grad keys is operational.

1.2 DISPLAY

The display contains a 12 digit green digitron tube. This is divided into three parts, as shown below.

9 1.2345678-90

ENT ANS

The first digit indicates the number of the data register used for storage when the ENT lamp is alight, or the number of the data register used for recall when the ANS lamp is alight. The 10 digits on the right are used for a 10 digit floating point number, or an 8 digit mantissa and 2 digit exponent for a scientific notation number (if a negative number is used in scientific notation only a 7 digit mantissa can be used). In some calculations a "–" sign appears momentarily while the machine is working. It is not possible to use the keyboard whilst this happens.

1.3 ARITHMETIC FUNCTIONS

[+] [–] [÷] [X] [EXP] [1/x] [π] [√] [+/–]

The basic four functions are entered algebraically. This means that a problem is entered in the same order that it is written down.

EXAMPLE Calculate 1 + 2 + 3 + 4 + 5

Key	Display	Comment
1	1.	
[+]	1.	
2	2.	
[+]	3.	Partial Answer
3	3.	
[+]	6.	Partial Answer
4	4.	
[+]	10.	Partial Answer
5	5.	
[=]	15.	Total

Notice that a partial answer was displayed after each operator key was used. The [=] key ended the calculation. The other three basic keys are used in the same way. They may also be used for chain calculations.

EXAMPLE Calculate a) (23 x 59) – 10 b) 123 ÷ 567 x 89

	Key	Display	Comment
a)	23	23.	
	[X]	23.	
	59	59.	
	[–]	1357.	Partial Answer
	10	10.	
	[=]	1347.	Answer
b)	123	123.	
	[÷]	123.	
	567	567.	
	[X]	0.216931216	Partial Answer
	89	89.	
	[=]	19.30687831	Answer

When entering a number in scientific notation the following keying operation is used.

<u>EXAMPLE</u> *Calculate $1.23 \times 10^6 - 3.45 \times 10^4$*

Key	Display		Comment
1.23	1.23		
[exp]	1.23	00	
6	1.23	06	1.23×10^6
[–]	123000.0		number less than 10^{10} therefore displayed in floating point.
3.45	3.45		
[exp]	3.45	00	
4	3.45	04	3.45×10^4
[=]	1195500.		Answer

N.B. The [exp] key does not function if the mantissa contains more than 8 digits (7 if negative).

The calculator will automatically show a "–" sign for a negative number. To enter a negative number the [+/–] key is used.

<u>EXAMPLE</u> *Calculate $(-1.234 \times 10^{-56}) \times (3.45 \times 10^{23})$*

Key	Display		Comment
1.234	1.234		
[+/–]	–1.234		negative mantissa
[exp]	–1.234	00	
56	–1.234	56	
[+/–]	–1.234	–56	negative exponent
[X]	–1.234	–56	
3.45	3.45		
[exp]	3.45	00	
23	3.45	23	
[=]	–4.2573	–33	Answer

N.B. The [+/–] key *must* be used *after* the mantissa is entered. It can also be used to change the sign of a partial answer.

Clear Keys

The clear key, $\boxed{C}$, clears the keyboard entry if a correction is necessary. Any value shown in the display, providing it is an answer to a function and not a partial answer, will also be cleared. The all clear key, $\boxed{AC}$, clears both working registers, but not the independent memory or the data registers.

EXAMPLE Calculate (25 x 3) + (6)

Key	Display	Comment
25	25.	
$\boxed{\times}$	25.	
3	3.	
$\boxed{+}$	75.	partial answer
7	7.	'7' pressed by mistake
$\boxed{C}$	0.	clear '7'
6	6.	
$\boxed{=}$	81.	Answer

The FX201P has an automatic constant facility. This is activated by a double keying of the constant function needed.

EXAMPLE Calculate:

a) 1.4 + 2.4, 1.5 + 2.4, 1.6 + 2.4, 1.7 + 2.4
b) 2 − 1.6, 3 − 1.6, 4 − 1.6
c) 1.8 x 1.9, 1.7 x 1.9, 1.6 x 1.9
d) 20 ÷ 1.7, 30 ÷ 1.7, 40 ÷ 1.7
e) $1.9^2, 1.9^3, 1.9^4$

	Key	Display	Comment
a)	2.4	2.4	
	$\boxed{+}\boxed{+}$	2.4	Activate constant +
	1.4	1.4	
	$\boxed{=}$	3.8	Ans 1.
	1.5	1.5	
	$\boxed{=}$	3.9	Ans 2.
	1.6	1.6	
	$\boxed{=}$	4.	Ans 3.
	1.7	1.7	
	$\boxed{=}$	4.1	Ans 4.

	Key	Display	Comment
b)	1.6	1.6	
	[−] [−]	1.6	Activate constant −
	2	2.	
	[=]	0.4	Ans 1.
	3	3.	
	[=]	1.4	Ans 2.
	4	4.	
	[=]	2.4	Ans 3.
c)	1.9	1.9	
	[×] [×]	1.9	Activate constant
	1.8	1.8	
	[=]	3.42	Ans 1.
	1.7	1.7	
	[=]	3.23	Ans 2.
	1.6	1.6	
	[=]	3.04	Ans 3.
d)	1.7	1.7	
	[÷] [÷]	1.7	Activate constant
	20	20.	
	[=]	11.76470588	Ans 1.
	30	30.	
	[=]	17.64705882	Ans 2.
	40	40.	
	[=]	23.52941176	Ans 3.
e)	1.9	1.9	
	[×] [×]	1.9	Activate constant
	[=]	3.61	Ans 1.
	[=]	6.859	Ans 2.
	[=]	13.0321	Ans 3.

Reciprocal calculations are made using the [1/x] key.

EXAMPLE Calculate $\frac{1}{5 + \frac{1}{3}}$

Key	Display	Comment
5	5.	
[+]	5.	
3	3.	
[1/x]	0.333333333	($\frac{1}{3}$)
[=]	5.333333333	($5 + \frac{1}{3}$)
[1/x]	0.1875	Answer

The square root of the displayed number is extracted using the $\boxed{\sqrt{\ }}$ key.

EXAMPLE Calculate $\sqrt{2} + \sqrt{3}$

Key	Display	Comment
2	2.	
$\boxed{\sqrt{\ }}$	1.414213562	$\sqrt{2}$
$\boxed{+}$	1.414213562	
3	3.	
$\boxed{\sqrt{\ }}$	1.732050807	$\sqrt{3}$
$\boxed{=}$	3.146264369	Answer

The constant, pi, is entered in the display, to 10 digits when the $\boxed{\pi}$ key is used. This replaces the contents of the display. Any pending operations are lost.

1.4 MEMORY FUNCTIONS

$\boxed{\text{M+}}$ $\boxed{\text{MR}}$ $\boxed{\text{MC}}$ $\boxed{\text{ENT}}$ $\boxed{\text{ANS}}$

The calculator has one independent memory and ten addressable data registers. The independent memory is accessed using the $\boxed{\text{M+}}$ key — the data registers by the $\boxed{\text{ENT}}$ key. If the $\boxed{\text{M+}}$ key is used to terminate a calculation the final answer is accumulated into the independent memory. Numbers may be subtracted from the memory by additional use of the $\boxed{+/-}$ key. The independent memory is cleared by the $\boxed{\text{MC}}$ key.

EXAMPLE Calculate a) (123 x 456) – (78 x 9)
b) (20 + 30 + 40) x (50 + 60 + 70)

	Key	Display	Comment
a)	123	123.	
	$\boxed{\times}$	123.	
	456	456.	
	$\boxed{\text{M+}}$	56088.	Partial answer entered in memory.
	78	78.	
	$\boxed{\times}$	78.	
	9	9.	
	$\boxed{+/-}$	–9.	
	$\boxed{\text{M+}}$	–702.	Partial answer added to memory.
	$\boxed{\text{MR}}$	55386.	Answer.
	$\boxed{\text{MC}}$		Clear memory.
b)	20	20.	
	$\boxed{+}$	20.	
	30	30.	
	$\boxed{+}$	50.	
	40	40.	
	$\boxed{\text{M+}}$	90.	Partial answer entered in memory.
	50	50.	
	$\boxed{+}$	50.	
	60	60.	
	$\boxed{+}$	110.	
	70	70.	
	$\boxed{\times}$	180.	Partial answer.
	$\boxed{\text{MR}}$	90.	Recall from memory.
	$\boxed{=}$	16200.	Answer.

Use of the data registers is required when partial answers are needed for further calculations.

EXAMPLE Calculate $\frac{(9 \times 6) + 3}{(7 - 2) \times 8}$

	Key		Display	Comment
	9		9.	
	[×]		9.	
	6		6.	
	[+]		54.	
	3		3.	
	[=]		57.	
[ENT]	1	(ENT) 1	57.	57 entered in register 1.
	7		7.	
	[−]		7.	
	2		2.	
	[×]		5.	
	8		8.	
	[=]		40.	
[ENT]	2	(ENT) 2	40.	40 entered in register 2.
[ANS]	1	(ANS) 1	57.	Recall 1.
	[÷]		57.	
[ANS]	2	(ANS) 2	40.	Recall 2.
	[=]		1.425	

(This example could have been worked using the [M+] and [MR] keys — but the above method shows the use of the data registers). When a number is entered into register any previously held number is cleared.

1.5 TRIGONOMETRIC FUNCTIONS

$\boxed{\text{sin}}$ $\boxed{\text{cos}}$ $\boxed{\text{tan}}$ $\boxed{\text{arc}}$ $\boxed{{}^{\circ}{}'{}''}$

The $\boxed{\text{sin}}$, $\boxed{\text{cos}}$, and $\boxed{\text{tan}}$ keys obtain each trigonometric value of the number displayed. The mode of calculation, radians, degrees or grads, is indicated by the position of the angular mode selector. All entries must be in decimal format, although use of the sexagesimal → decimal conversion key, $\boxed{{}^{\circ}{}'{}''}$, allows entry of degrees in degrees, minutes and seconds format.

EXAMPLE Calculate:

a) sin 63° 52' 41"

b) cos $\frac{\pi}{3}$ rads.

c) tan (–35) grads.

d) 2 sin 45° x cos 65°

	Key	Display	Comment
	Select "DEG"		
a)	63	63.	Enter degrees.
	$\boxed{{}^{\circ}{}'{}''}$	63.	
	52	52.	Enter minutes.
	$\boxed{{}^{\circ}{}'{}''}$	63.86666666	63° 52' in decimal format.
	41	41.	Enter seconds.
	$\boxed{{}^{\circ}{}'{}''}$	63.87805555	63° 52' 41" in decimal format.
	$\boxed{\text{sin}}$	0.89785901	Answer.
b)	Select "RAD"		
	$\boxed{\pi}$	3.141592653	
	$\boxed{\div}$	3.141592653	
	3	3.	
	$\boxed{=}$	1.047197551	$\frac{\pi}{3}$
	$\boxed{\text{cos}}$	0.5	Answer.
c)	Select "GRAD"		
	35	35.	
	$\boxed{+/-}$	–35.	
	$\boxed{\text{tan}}$	–0.61280079	Answer.
d)	Select "DEG"		
	2	2.	
	$\boxed{\times}$	2.	
	45	45.	
	$\boxed{\text{sin}}$	0.70710678	sin 45°
	$\boxed{\times}$	1.41421356	2 sin 45°
	65	65.	
	$\boxed{\text{cos}}$	0.42261826	cos 65°
	$\boxed{=}$	0.597672473	Answer

Inverse trigonometric functions may be obtained by using the $\boxed{\text{arc}}$ key before the appropriate function.

EXAMPLE Calculate:
a) arc sin 0.5 in degrees.
b) arc cos 0.6 in rads.
c) arc tan 20 in grads.
d) arc sin 0.7 – arc cos 0.6 in degrees.

	Key	Display	Comment
a)	Select "DEG"		
	0.5	0.5	
	$\boxed{\text{arc}}\boxed{\text{sin}}$	30.	Answer
b)	Select "RAD"		
	0.6	0.6	
	$\boxed{\text{arc}}\boxed{\text{cos}}$	0.92729522	Answer
c)	Select "GRAD"		
	20	20.	
	$\boxed{\text{arc}}\boxed{\text{tan}}$	96.81955	Answer
d)	Select "DEG"		
	0.7	0.7	
	$\boxed{\text{arc}}\boxed{\text{sin}}$	44.427004	arc sin 0.7
	$\boxed{-}$	44.427004	
	0.6	0.6	
	$\boxed{\text{arc}}\boxed{\text{cos}}$	53.130102	arc cos 0.6
	$\boxed{=}$	–8.703098	Answer

It is possible to convert from one angular mode to another – although indirectly.

EXAMPLE Convert 25° to radians

Key	Display	Comment
Select "DEG"		
25	25.	
$\boxed{\text{sin}}$	0.42261826	sin 25°
Select "RAD"		
$\boxed{\text{arc}}\boxed{\text{sin}}$	0.43633231	Answer

1.6 LOGARITHMIC FUNCTIONS

$\boxed{\ln}$ $\boxed{\log}$ $\boxed{e^x}$ $\boxed{10^x}$ $\boxed{y^x}$

Logarithms may be calculated to base '10' or to base 'e'. The $\boxed{\log}$ key calculates the common log of the number displayed. The $\boxed{\ln}$ key calculates the natural log. Antilogarithms are obtained by use of the $\boxed{10^x}$ or $\boxed{e^x}$ keys. A displayed number, x, may be raised to the power y by use of the $\boxed{x^y}$ key.

EXAMPLE Calculate:
a) $10^{1.23}$
b) $e^{4.5}$
c) log 123
d) ln 90
e) $2.3^{5.6}$

	Key	Display	Comment
a)	1.23	1.23	
	$\boxed{10^x}$	16.982437	Answer
b)	4.5	4.5	
	$\boxed{e^x}$	90.017131	Answer
c)	123	123.	
	$\boxed{\log}$	2.0899051	Answer
d)	90	90.	
	$\boxed{\ln}$	4.4998097	Answer
e)	2.3	2.3	x
	$\boxed{x^y}$	2.3	
	5.6	5.6	y
	$\boxed{=}$	106.09035	Answer

These functions may be chained — but x^y must be used as the first term in an expression only.

EXAMPLE Calculate:
a) log 456 ÷ log 789
b) $e^2 + e^3$
c) $2^3, 3^3, 4^3$

	Key	Display	Comment
a)	456	456.	
	$\boxed{\log}$	2.6589648	
	$\boxed{\div}$	2.6589648	
	789	789.	
	$\boxed{\log}$	2.897077	
	$\boxed{=}$	0.917809502	Answer

b)	2	2.	
	$\boxed{e^x}$	7.3890561	
	$\boxed{+}$	7.3890561	
	3	3.	
	$\boxed{e^x}$	20.085537	
	$\boxed{=}$	27.4745931	Answer
c)	3	3.	
	$\boxed{x^y}\boxed{x^y}$	3.	enter x^3 as a constant
	2	2.	
	$\boxed{=}$	8.	Answer
	3	3.	
	$\boxed{=}$	27.	Answer
	4	4.	
	$\boxed{=}$	64.	Answer

1.7 ACCURACY, INPUT RANGES AND ERROR FUNCTIONS

1.7.1. Accuracy

For most functions the calculator is accurate to ± 1 in the 8th digit. For $\sqrt{x}$, $\frac{1}{x}$, and .... the accuracy is normally ± 1 in the 10th digit. For x^y the accuracy is only ± 1 in the 7th digit. The accuracy tends to be worse towards either end of the input range for a particular function.

1.7.2. Input ranges

sin x, cos x, tan x	$\lvert x \rvert \leqslant 1440^\circ$, 8π rads, 1600 grad.
arc sin x, arc cos x	$\lvert x \rvert \leqslant 1$
arc tan x	$\lvert x \rvert < 1 \times 10^{100}$
log x, ln x	$0 < x < 1 \times 10^{100}$
10^x	$\lvert x \rvert < 100$
e^x	$-227 < x < 230$
x^y	$0 < x < 1 \times 10^{100}$
$\sqrt{x}$	$0 \leqslant x \leqslant 1 \times 10^{100}$
$\frac{1}{x}$	$\lvert x \rvert < 1 \times 10^{100}$ $x \neq 0$
....	Up to a second.

1.7.3. Error indications

Overflow is indicated by an 'E' sign and stops further calculations. To release the keyboard depress the [AC] key. All previous results in the working registers are lost.

Overflow occurs:

a) When an answer or accumulated total in the independent memory exceeds 1×10^{100}

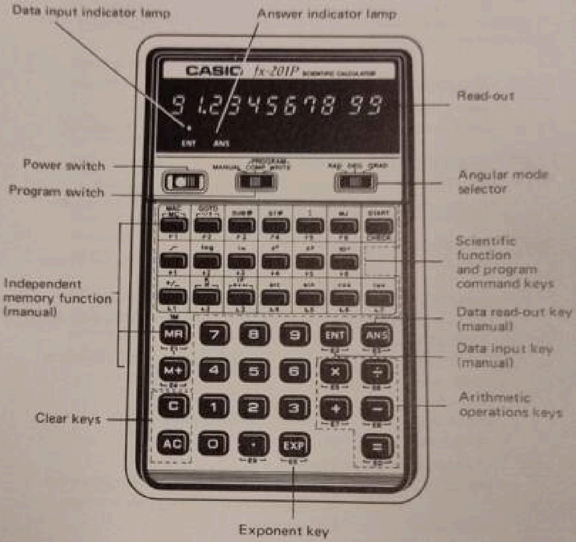
or b) When the function calculations are performed on a number exceeding their input range.

N.B. The contents of the independent memory are protected against overflow and the total accumulated up to the point of the overflow, calculation is recalled by the [MR] key after the overflow check is released by the [AC] key.

Part 2
PROGRAM CALCULATIONS

2.1 KEYBOARD – PROGRAMMING FUNCTIONS

The following keys are operational during the writing of programs, in addition to those mentioned in the manual operating section.



GOTO	Unconditional jump key
SUB #	Subroutine key
ST #	Statement number key
;	Statement end key
CHECK	Check key
K	Enter constant key
IF	Conditional jump key
IM	Indirect memory key
I	Indirect memory access key
MJ	Manual jump key
MAC	Clear all memories

The following keys perform different operations in the program mode.

$\frac{1}{x}$, π , ... , M+ , M− , MR , MC , C , AC

When the program switch is set to the COMPute position the keyboard is used to perform calculations using a program.

In the **WRITE** mode the keyboard is used to store a program in the calculator, or to check a stored program.

As some keys have different functions for **COMPute** or **WRITE** the following list is a summary of each key function

[ENT] **DATA ENTRY KEY:**

WRITE mode . . . . . . Use to write in data input messages.

COMP mode . . . . . . Use to advance the program by the operation Data [ENT] .

[ANS] **ANSWER KEY:**

WRITE mode . . . . . . Use to write in answer display messages.

COMP mode . . . . . . Use after reading out an answer to advance the program.

[MAC] **MEMORY ALL CLEAR KEY ([MC]) :**

WRITE mode . . . . . . Use to write in a clear command for the 10 data memories and I-memory.

COMP mode . . . . . . Clears 10 data memories and I-memory.

It works as independent memory clear (MC) only in the **MANUAL** mode.

[GOTO] **GOTO KEY ([1/x]) :**

WRITE mode . . . . . . Use to write in unconditional jump commands.

COMP mode . . . . . . Works as [1/x] (reciprocal key).

[SUB #] **SUBROUTINE KEY:**

WRITE mode . . . . . . Use to write in subroutine programs.

COMP mode . . . . . . No command.

[ST #] **STATEMENT NUMBER KEY:**

WRITE mode . . . . . . Use to write in the address to which both conditional and unconditional jumps are made.

COMP mode . . . . . . No command.

[:] **MESSAGE END KEY:**

WRITE mode . . . . . . Use to divide formulae and messages in programming.

COMP mode . . . . . . No command.

[MJ] **MANUAL JUMP KEY:**

WRITE mode . . . . . . Use to write in MJ commands in programs.

COMP mode . . . . . . Use during execution of a program to make a jump at the MJ position in the program.

[CHECK] **CHECK KEY ([START] START KEY):**

WRITE mode . . . . . . Use when advancing a written program ahead 1 step (called program check). Shown as [CHK] in this manual.

COMP mode . . . . . . Use to start a program calculation (to read the program from the first step). Shown as [START] in this manual.

K **CONSTANT WRITE-IN KEY (π):**

WRITE mode . . . . . . Use to write in constants in a program.

COMP mode . . . . . . Works as a π (Pi) key.

IF **IF KEY (° ' "):**

WRITE mode . . . . . Use to write in conditional jump commands.

COMP mode . . . . . . Works as a ° ' " (Sexagesimal →decimal conversion key.)

I **INDIRECT KEY (M+):**

WRITE mode . . . . . Use to write in the command to store a value in the I-memory.

COMP mode . . . . . . No command.

IM **INDIRECT MEMORY KEY (MR):**

WRITE mode . . . . . Use to write in the command to indirectly address the values stored in the I-memory during calculation.

COMP mode . . . . . . No command.

It works as MR (Independent memory recall) only in MANUAL mode.

C **CLEAR KEY:**

WRITE mode . . . . . Use to back up a written program one step and clear.

COMP mode . . . . . . Use to clear displayed data or answers.

AC **ALL CLEAR KEY:**

WRITE mode . . . . . Use to erase a written program.

COMP mode . . . . . . Use when desired to stop a program calculation.

2.2 DISPLAY

When the calculator is in the **WRITE** mode, the display is used to indicate the step programmed from the keyboard. The code, in brown, under each key is repeated in the display. The code of the current step, and of the two previous steps are shown. The three figure number on the far left is the step number.

EXAMPLE			
097	E2	6	Γ5
Step No.	*Code for Step 095*	*Code for Step 096*	*Code for Step 097*

The above example shows that step 097 is a statement end [:] . The previous step, 096, was the number 6 and step 095 was an [ENT] command. Hence we can see that the program steps were [ENT] 6:

When the calculator is in the **COMP**ute mode, the number appearing on the far left of the display indicates a storage register. If the (ENT) lamp is alight a number must be entered in that register and if the (ANS) lamp is alight, the number displayed is the contents of the register indicated.

EXAMPLE

6 1.567
(ANS)

This indicates that an answer in register 6 has been displayed, its value being 1.567.

2.3 PROGRAM STEPS AND COMMAND CODES

The FX201P has 127 steps available for programming. Each key stroke in the **WRITE** mode is counted as one step. Step 001 is the first available for a program step, step 000 being unassignable. The command codes for each key are those indicated below each key. Three keys have no codes. In the **WRITE** mode the [C] key back-steps the program for editing, while the [AC] key clears the total program. The [START] or check key has no command associated with it.

2.4 BASIC PROGRAMMING

The FX20IP has a unique, but powerful, programming "language". Each variable in the formula to be programmed is allocated a data register, i.e. labeled. The appropriate mathematical manipulation is then carried out on these labeled variables. Consequently, numbers used as constants have to be labeled as constants. For example, if two numbers, a and b, were to be added, number a could be allocated label 1 and number b could be allocated label 2. If the answer, c, were to be allocated label 3, then the calculator would be programmed to solve

$$3 = 1 + 2 \quad \text{i.e.} \quad c = a + b \,.$$

All of the keyboard functions, but not $\boxed{\frac{1}{x}}$, $\boxed{\pi}$ or $\boxed{}$, are available for use. As the calculator does not inherently know what labels we are going to use for input data these have to be indicated at the very beginning of the program. Similarly the labels used for the answers have to be indicated at the end of a program. As an example we shall try to multiply three numbers.

EXAMPLE Calculate a x b x c

Key	Comment
Select **WRITE** mode.	
ENT 1 : 2 : 3 :	Variables a, b and c labeled and entered.
4 = 1 x 2 x 3 :	The answer, labeled 4, equals a x b x c.
ANS 4 :	Answer, labeled 4, required.

If we now wish to calculate 124 x 356 x 790 switch the **COMP**ute mode. Depress START key.

Key	Display		Comment
	1 ENT	0.	Calculator is requesting value of variable 1.
124		124.	
ENT			Enter 124
	2 ENT	0.	Calculator requests variable 2.
356		356.	
ENT			Enter 356
	3 ENT	0.	Calculator requests variable 3.
790		790.	
ENT			Enter 790
	4 ANS	34873760.	Variable 4 i.e. answer.

This is a rather simple example, but shows the method of entering variables and arithmetic commands.

Further examples will illustrate the use of other functions.

If two calculations can be made with the same data, then two variables may be labeled to give two answers.

EXAMPLE Calculate the surface area and volume of a regular octahedron, with sides a) of 10cm b) of 7cm and c) of 15cm.

The surface area, $S = a^2\, 2\sqrt{3}$ and the volume $V = \frac{a^3\sqrt{2}}{3}$ where a is the side length.

Label a as variable ①
S as variable ②
V as variable ③

Hence ② = 2√3 × ① × ① $(S = a^2\, 2\sqrt{3})$

and ③ = $\frac{\sqrt{2}}{3}$ × ① × ① × ① . $(V = \frac{a^3\sqrt{2}}{3})$

The program will therefore be:

ENT 1:	variable "a" labeled 1.
2 = K2 × K3 √ × 1 × 1:	calculate S or label 2.
3 = 1 x^y K3 × K2√ ÷ K3:	calculate V or label 3.
ANS 2: 3:	Display 2 and 3.

To write in this program, set the program switch to **WRITE** and key in the program as shown above. To perform the calculation, using the program, set the program switch to **COMPute.** Then perform the following operations.

Key	Display		Comment
START	1 ENT	0.	Request for variable 'a'
10	1 ENT	10.	a_1 — first value
ENT	2 ANS	346.4101614	Answer S_1
ANS	3 ANS	471.4045206	Answer V_1
START	1 ENT	0.	Request a
7	1 ENT	7.	Second value of a_2
ENT	2 ANS	169.740979	Answer S_2
ANS	3 ANS	161.6917505	Answer V_2
START	1 ENT	0.	Request a
15	1 ENT	15.	Third value of a_3
ENT	2 ANS	779.4228631	Answer S_3
ANS	3 ANS	1590.990257	Answer V_3

N.B. Once the START key has been used, and the program initiated, it does not matter whether the ANS or ENT keys are used. The program is advanced using either key.

Checking and editing programs

To illustrate the procedure for checking and editing suppose that a similar calculation to the previous one is required, except that the surface area and volume of a regular tetrahedron are involved. In this case the formulae used are very similar.

Surface area, $S = \sqrt{3}\ a^2$

and Volume, $V = \dfrac{\sqrt{2}\ a^3}{12}$

Instead of rewriting the program and reprogramming the calculator it is only necessary to change a few constants in the original program.

EXAMPLE Calculate the surface area and volume of a regular tetrahedron with sides of a) 10cm b) 7cm and c) 15cm

Label 'a' as variable 1

'S' as variable 2 where $S = \sqrt{3}\ a^2$

and 'V' as variable 3 where $V = \dfrac{\sqrt{2}\ a^3}{12}$

The program will therefore be:

ENT 1:	variable 'a' labeled 1
2 = K3$\sqrt{\ }$ x 1 x 1:	"S"
3 = 1 x^Y K3 x K2 $\sqrt{\ }$ ÷ K12:	"V"
ANS 2: 3:	

Program 'CHECK' recalls the written program into the display, thus confirming the contents of the program. Each time the $\boxed{\text{CHECK}}$ key is used in the WRITE mode, the step numbers and contents are displayed one at a time.

If the two programs, the previous one and the one above, are compared, the difference between them can be easily seen.

a) the 'K2x' has to be erased from the second line of the first program,

and b) the last term in the third line, 'K3', should be changed to 'K12'

To change the first program to the second one switch to **WRITE** mode: Steps 006, 007, and 008 have to be erased. This is achieved using the C key as follows:

Key	Display				Comment
CHK	001			E2	ENT
CHK	002		E2	1	1
CHK	003	E2	1	┌5	:
CHK	004	1	┌5	2	2
CHK	005	┌5	2	E0	=
CHK	006	2	E0	L2	K (to be erased)
C	005	┌5	2	E0	program backed up to 005
CHK	006	2	E0	00	no command
CHK	007	E0	00	2	2 (to be erased)
CHK	008	00	2	E5	x (to be erased)
C	007	E0	00	2	back up and clear
C	006	2	E0	00	back up and clear
CHK	007	E0	00	00	no command
CHK	008	00	00	00	no command

Hence steps 006, 007 and 008 have been erased from the program. The program will now automatically jump from 005 to 009 during operation.

Now advance the program to step 028

CHK	028	⊢1	E6	L2	K

The program must be rewritten from this point as an extra step has to be introduced.

1	029	E6	L2	1	1
2	030	L2	1	2	2
:	031	1	2	┌5	:
ANS	032	2	┌5	E3	ANS
2	033	┌5	E3	2	2
:	034	E3	2	┌5	:
3	035	2	┌5	3	3
:	036	┌5	3	┌5	:

The program has now been *rewritten* for the tetrahedron problem, and may be solved in the usual way – set mode to **COMP**ute

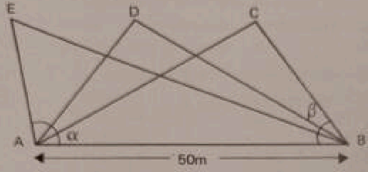
START	1 (ENT)	0.	Enter first value of a
10	1 (ENT)	10.	a_1
ENT	2 (ANS)	173.2050807	S_1
ANS	3 (ANS)	117.8511301	V_1
	and so on.		

Using the AC key in the **WRITE** mode will clear **all** program storage registers and set the program back to step 000.

2.5 MANUAL JUMP $\boxed{\text{MJ}}$

The manual jump function allows the user to control a jump operation from the keyboard during **COMP**ute mode. This is useful for successive calculations where data is common to each one, or where the results of successive calculations have to be totaled. (The use of the $\boxed{\text{START}}$ key clears all memory registers).

EXAMPLE *Three triangles are on a common base AB, as in the diagram. For given values of their base angles calculate their respective areas, and the total area of all three.*



△	α	β
ACB	41°	76°
ADB	61° 32′	49° 25′
AEB	87° 06′	37° 53′

$$\text{Area of triangle} = \frac{1}{2}\ AB^2\ \frac{\sin\alpha\ \sin\beta}{\sin(\alpha+\beta)}$$

Let AB — variable 1
α — variable 2
β — variable 3
Area — variable 4
Total Area — variable 5

The program then is:

MAC:	
ENT 1:	enter AB.
MJ	
ENT: 2: 3:	enter α, β
6 = 2 + 3	Calculate α + β
4 = 1 x 1 x 2 sin x 3 sin ÷ 6 sin ÷ K2:	Calculate area.
5 = 5 + 4:	Sum areas.
ANS 4: 5:	

During the operation of the program, depression of $\boxed{\text{MJ}}$ instead of $\boxed{\text{START}}$ will cause program to jump to the beginning, thus accumulating answers in memory 5, whilst using the value of AB entered only once.

Program calculation:

Key	Display		Comment
START	1 ENT	0.	
50	1 ENT	50.	Enter AB.
ENT	2 ENT	0.	
41	2 ENT	41.	Enter α_1
ENT	3 ENT	0.	
76	3 ENT	76.	Enter β_1
ENT	4 ANS	893.050508095	Area 1
MJ	2 ENT	0.	Jump to calculate area 2
61 ˚''' 32 ˚'''	2 ENT	61.53333333	Enter α_2
ENT	3 ENT	0.	
49 ˚''' 25 ˚'''	3 ENT	49.41666666	Enter β_2
ENT	4 ANS	893.621914	Area 2
MJ	2 ENT	0.	
87 ˚''' 06 ˚'''	2 ENT	87.1	Enter α_3
ENT	3 ENT	0.	
37 ˚''' 53 ˚'''	3 ENT	37.88333333	Enter β_3
ENT	4 ANS	935.6388665	Area 3
ANS	5 ANS	3056.104087	Total area.

N.B. Remember to set angular mode to 'DEG'.

2.6 UNCONDITIONAL JUMP [GOTO] [ST #]

If an unconditional jump is required in the program, this may be accomplished by use of the [GOTO] key. Up to 10 different points may be programmed using a number key and [ST #].

EXAMPLE Calculate the total price of the following

Product	No.	Unit Price
A	15	£300
B	10	£180
C	12	£250

The same calculation has to be performed three times i.e. No. x unit price, and then totaled. Instead of using a manual jump, as in 2.5, use an unconditional jump. This will involve less keying operations.

If			
	No. of items	–	variable 1
	Unit price	–	variable 2
	Total price per product	–	variable 3
	Total price	–	variable 4
	Total no. of items	–	variable 5

The program is:

	MAC	
ST # 1:	ENT 1: 2:	enter no. of items and unit price.
	3 = 1 x 2:	Total price per product.
	4 = 3 + 4:	Total price.
	5 = 1 + 5:	Total no. of items.
	ANS 3:	Display total price per product.
	GOTO 1:	Goto ST # 1 unless [MJ] keyed.
	MJ	
	ANS 4: 5:	Display total price and no. of items.

Program calculation:

Key	Display			Comment
$\boxed{\text{START}}$	1	ENT	0.	
15	1	ENT	15.	enter 1st number.
$\boxed{\text{ENT}}$	2	ENT	0.	
300	2	ENT	300.	enter 1st price.
$\boxed{\text{ENT}}$	3	ANS	4500.	1st sub-total
$\boxed{\text{ANS}}$	1	ENT	0.	
10	1	ENT	10.	enter 2nd number
$\boxed{\text{ENT}}$	2	ENT	0.	
180	2	ENT	180.	enter 2nd price.
$\boxed{\text{ENT}}$	3	ANS	1800.	2nd sub-total
$\boxed{\text{ANS}}$	1	ENT	0.	
12	1	ENT	12.	enter 3rd number.
$\boxed{\text{ENT}}$	2	ENT	0.	
250	2	ENT	250.	enter 3rd price.
$\boxed{\text{ENT}}$	3	ANS	3000.	3rd sub-total
$\boxed{\text{MJ}}$	4	ANS	9300.	Total.

2.7 CONDITIONAL JUMP $\boxed{\text{IF}}$

The use of the $\boxed{\text{IF}}$ key allows a variable to be compared to a constant, or to the contents of another memory register. The statement is completely inclusive of all possible tests, and allows the program to branch to three numbered statements depending on the result of the test.

Hence M = m: A: B: C: is the form of the test, where the program will:

go to statement A if $M < m$,
go to statement B if $M = m$,
and go to statement C if $M > m$.

It is possible for A, B or C to be the same statement.

EXAMPLE Calculate the roots of the following quadratic equations

a) $8x^2 + 6x + 1 = 0$
b) $2x^2 - 28x + 98 = 0$
c) $2x^2 + 26x + 89 = 0$

The formula used is: $x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

If $b^2 - 4a = D$ then D can be compared to 0.

Hence if a – variable 1
b – variable 2
so c – variable 3
and D – variable 4

let any real roots be labeled 5 and 6
any compound roots be labeled 7
and any complex roots be labeled 8 and 9 (9 being the imaginary part)

(As $\frac{1}{2a}$ is needed for each calculation, calculate 2a and store in 0)

The program is:

ST # 4:	ENT	1: 2: 3:	enter a, b, c.
	4 =	K4 x 1 x 3:	4ac
	4 =	2 x 2 – 4:	$b^2 - 4ac = D$
	0 =	K2 x 1:	Store 2a in 0
	IF 4 =	KO: 1: 2: 3:	Compare D to 0. If D < O goto ST #1 If D = O goto ST #2 If D > O goto ST #3

ST # 1: 8 = 2 +/− ÷ 0: real part of complex root
9 = 4 +/− √ ÷ 0: imaginary part of complex root
ANS 8:9:
GOTO 4: Return to beginning

ST # 2: 7 = 2 +/− ÷ 0: Compound root
ANS 7:
GOTO 4:

ST # 3: 5 = 2 +/− + 4 √ ÷ 0: real root 1
6 = 2 +/− − 4 √ ÷ 0: real root 2
ANS 5: 6:
GOTO 4:

Program calculation

	Key	Display		Comment
	START	1 ENT	0.	
a)	8	1 ENT	8.	enter a_1
	ENT	2 ENT	0.	
	6	2 ENT	6.	enter b_1
	ENT	3 ENT	0.	
	1	3 ENT	1.	enter c_1
	ENT	5 ANS	−0.25	x_1 real roots
	ANS	6 ANS	−0.5	x_2
b)	ENT	1 ENT	0.	
	2	1 ENT	2.	enter a_2
	ENT	2 ENT	0.	
	−28	2 ENT	−28.	enter b_2
	ENT	3 ENT	0.	
	98	3 ENT	98.	enter c_2
	ENT	7 ANS	7.	x_1 , x_2 — compound root
c)	ENT	1 ENT	0.	
	2	1 ENT	2.	enter a_3
	ENT	2 ENT	0.	
	26	2 ENT	26.	enter b_3
	ENT	3 ENT	0.	
	89	3 ENT	89.	enter c_3
	ENT	8 ANS	−6.5	real part complex roots
	ANS	9 ANS	1.5	imaginary part

Therefore the answers are a) $x_1 = -0.25$ $x_2 = -0.5$
b) $x_1 = x_2 = 7$
c) $x_1 = -6.5 + j1.5$, $x_2 = -6.5 - j1.5$

2.8 INDIRECT ADDRESSING $\boxed{\text{I}}$ $\boxed{\text{IM}}$

All previous calculations have involved the use of direct memory addressing e.g. ENT 1: 2: 3: Calculations have then been performed by calling on the memories directly. With indirect addressing it is possible to store a number, between 0 and 9, in the indirect memory and use this to perform calculations with.

For example if a program states that I = K5: this means that the indirect memory now has the number 5 stored in it. Hence a program instruction IM = 2 x 3: would be the same as an instruction 5 = 2 x 3:

It would not be good programming practice to use this example. A more usual use of indirect addressing is for sequentially addressing the memory registers. Thus a program step I = I + K1: will add one to the indirect memory each time a calculation is performed. If $\boxed{\text{IM}}$ is called in each calculation it will have a higher value. This is useful for counting.

EXAMPLE Add the following nine numbers together and store each number in a separate memory.
18, 27, 35, 42, 48, 52, 55, 57, 58.

One program would be:

```
ENT 1: 2: 3: 4: 5: 6: 7: 8: 9:
0 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9:
ANS 0:
```

If the indirect memory is used the program can be written as:

```
ST #1:  I = I + K1:
        ENT IM:
        0 = 0 + IM:
        GOTO 1:
        MJ
        ANS 0:
```

In this program the indirect memory is initially 1. Hence the first number will be put in memory 1 and added to 0. Next time round the loop the indirect memory is 2. Hence the second number will be put in memory 2 and also added to 0. This will continue until $\boxed{\text{MJ}}$ is keyed when the contents of 0 will be displayed.

The entry code for the indirect memory is the letter E. Thus a request for entry into the indirect memory is shown as E ENT.

An extension of the program above is used for classification.

EXAMPLE *There are nine classification codes, 1 to 9. The total for each code is required, and also the grand total.*

Code	Data		Program
3	1850		MAC
1	3100	ST #1:	ENT I: O:
2	2000		IM = IM + O:
9	3600		GOTO 1:
4	6120		MJ
2	1450		I = KO:
8	3880		O = KO:
5	2230	ST #2:	I = I + K1:
3	5360		O = O + IM:
5	4870		ANS IM:
6	3190		IF I = K9: 2: 3: 3:
7	2310	ST #3:	ANS O:
1	2500		
7	1960		
8	3300		
5	1250		
4	1890		

In this example, each time the code number is entered into the indirect memory this labels the next data being entered. Previous data from that memory is then added to the newly entered data and the total is put back. Once all the data has been entered a manual jump causes the two registers to be cleared, and the program goes into a counting routine bringing the contents of each memory to the display. Once the contents of memory 9 have been displayed the total number, in memory O, is displayed.

Program operation:

Key	Display			Comment
$\boxed{\text{START}}$	O	ENT	0.	
1850	O	ENT	1850.	enter first data
$\boxed{\text{ENT}}$	E	ENT	0.	
3	E	ENT	3.	enter first code
$\boxed{\text{ENT}}$	O	ENT	0.	
3100	O	ENT	3100.	enter second data
$\boxed{\text{ENT}}$	E	ENT	0.	
1	E	ENT	1.	enter second code

1890	O (ENT)	1890.	enter last data
ENT	E (ENT)	0.	
4	E (ENT)	4.	enter last code
ENT	O (ENT)	0.	
MJ	1 (ANS)	5600.	first total
ANS	2 (ANS)	3450.	second total
ANS	3 (ANS)	7210.	third total
ANS	4 (ANS)	8010.	fourth total
ANS	5 (ANS)	8350.	fifth total
ANS	6 (ANS)	3190.	sixth total
ANS	7 (ANS)	4270.	seventh total
ANS	8 (ANS)	7180.	eighth total
ANS	9 (ANS)	3600.	ninth total
ANS	O (ANS)	50860.	grand total.

2.9 SUBROUTINES $\boxed{\text{SUB \#}}$

If a formula, or part of a program, is repeated many times in the same program it is possible to assemble it as a subroutine. It can then be called on when required, thus reducing the number of program steps.

<u>EXAMPLE</u> *Permutations and combinations require the use of factorial numbers. It is possible to write a subroutine to solve factorials and use this in a calculation e.g. How many combinations of 5 cards can be removed from a pack of 52?*

$$C^n_m = \frac{n\,!}{m\,!\,(n-m)\,!} \quad \text{where } n = 52,\ m = 5$$

The subroutine for n ! is:

```
SUB #O :  9 = K1:
ST #1  :  9 = 9 x 8:
          8 = 8 - K1:
        IF 8 = K1: 2: 2: 1:
ST #2  :  etc.
```

In this subroutine memory 9 is set to number 1, and memory 8 contains the number, n, requiring a factorial. Statement number 1 calculates n x 1, then subtracts 1 from n, tests to see if this value equals 1. If it is greater than 1 then n x (n – 1) is calculated. When memory 8 equals 1 the calculation stops and continues with the rest of the program, statement 2. Statement 2 is the next step in the program after the "goto subroutine" command.

The program for C^n_m is then:

MAC	
ENT 1: 2:	Enter n, m
8 = 1:	8 = n
GOTO O:	Goto Subroutine O, calculate n !
4 = 9:	4 = n !
8 = 2:	8 = m
GOTO O:	Goto Subroutine O, calculate m !
5 = 9:	5 = m !
8 = 1 – 2:	8 = n – m
GOTO O:	Goto Subroutine O, calculate (n – m) !
6 = 4 ÷ 5 ÷ 9:	C^n_m
ANS 6:	<u>Answer.</u>

```
SUB #O  : 9 = K1:
ST #1   : 9 = 9 x 8:                    Subroutine
            8 = 8 - K1:
        IF 8 = K1: 2: 2: 1:
ST #2   :
```

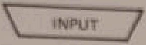
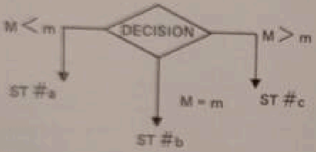
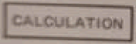
Program operation:

Key	Display		Comment
$\boxed{\text{START}}$	1 ENT	0.	
52	1 ENT	52.	enter n
$\boxed{\text{ENT}}$	2 ENT	0.	
5	2 ENT	5.	enter m
$\boxed{\text{ENT}}$	6 ANS	2598959.999	C^n_m – answer.

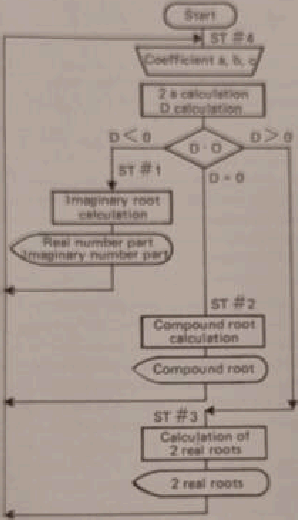
Therefore the answer is 2598960.

2.10 FLOW CHARTS

It is sometimes easier, where complicated programs are concerned, to arrange the logical steps of the program diagramatically. This is called a flow chart. The various program functions that have already been discussed can be given different symbols – shown below.

Symbol	Program function
	START
	ENT a: b: c:
	IF M – m: a: b: c:
	e.g. 1 – 2 + 3:
	Unconditional jump GOTO n:
	Jump to here ST #n
	ANS 1: 2:
	

To illustrate the use of a flow chart, consider the program for solving quadratic equations. This is repeated below, with the accompanying program.




```
ST #4
      ENT 1: 2: 3:

      4 = K4 × 1 × 3:
      4 = 2 × 2 − 4:
      0 = K2 × 1:

      IF 4 = K0: 1: 2: 3:
ST #1:
      8 = 2 +/− ÷0:
      9 = 4 +/− √ ÷0:

      ANS: 8: 9:

      GOTO 4:
ST #2:
      7 = 2 +/− ÷0:


      ANS 7:

      GOTO 4:
ST #3:
      5 = 2 +/− + 4 √ ÷0:
      6 = 2 +/− − 4 √ ÷ 0:
      ANS 5: 6:

      GOTO 4:
```

2.11 REVIEW OF PROGRAM FUNCTIONS

To write a program into the calculator

1. Set the program switch at **WRITE**.
2. Key-in the program in correct sequence.

(To erase a previously stored program, press [AC] after setting at **WRITE**.)

To perform a calculation using the program:

1. Set the program switch at **COMP**ute position.
   (When using a program including trigonometric or inverse trigonometric functions you must also set the angular mode selector as required.)
2. Press the [START] key.
3. When the (ENT) lamp lights, input the required data for the indicated memory numbers and press [ENT].
4. The answer is displayed by the memory number when the (ANS) lamp lights.
   (After reading out the answer, press [ANS] or [ENT] to advance the program.)
5. Press the [START] if desired to repeat the program calculation.
6. Press [AC] to stop the program calculation.

Making a program check:

1. Set the program switch at **WRITE**.
2. Each time [CHK] is depressed, step numbers and command codes are displayed for confirmation. (Steps containing no command are displayed as "00" or blank.)

* In order to make an addition to a program already written in, erase the command in the step where the addition is to be made and write in the new command by pressing the proper keys. Previous programs can be used when steps are erased or the number of steps is reduced but when the number of steps is increased the point from where the addition starts to the end must be written in again.

How to erase and change programs:

1. Advance to the required steps using program check.
2. Display the command to be erased and press [C].
   (Using [C], at the time the program is backed up the command that leaves the display is erased and becomes no command.)
3. Display the step just before the one to be changed and then write in the altered program.
4. For corrections, display the step just before the one to be corrected and write in the new command.

* Program calculations are not affected even if erased steps are in the middle of a program.

[MJ]

1. If [MJ] is pressed when a program is stopped (with (ENT) (ANS)) while performing a calculation, a jump is made up to the written in [MJ] (both backward and forward).
2. When more than one [MJ] is written in, the jump is made to the final one and the others are ignored.
3. If [MJ] is pressed when [MJ] is not written into the program, the program will not operate correctly.

How to perform totalling calculations

1. Prepare memory M for use in totaling. M = M + m: or M = m + M: is used. (m is the data or answer memory.)
2. This totaling use memory must be made 0 before starting the totaling.
3. To make the totaling memory 0.
   a. Do not include a clear command in the program but press the [MAC] key before entering the first data of the calculation.
   b. A clear command such as [MAC] , etc., is included in the first part of the program and below the 2nd line [MAC] is not read. There are also methods using [MJ] or [GOTO] ([GOTO] is explained below.).

[GOTO] [ST #]

1. By putting in GOTO N: a jump can be made to the program ST #N:.
2. N is a natural number from 1 to 9, 0.
3. GOTO N:' and ST #N: can be added at any position in the program. A maximum of 10 jumps can be used in accordance with the N number.
4. GOTO N1: is effective no matter how many times used but ST #N1: can only be used once; when ST # N1: is used more than once, only the last is effective.)
5. When there is no ST #N1: to correspond to GOTO N1: the program calculation is stopped.

[IF]

1. By programming IF M = m: A: B: C: , the place to which a jump is to be made can be made by comparing M and m. If M<m the jump is to ST #A; if M = m the jump is to ST #B; if M>m the jump is to ST #C.
2. M and m can be data memory numbers, I, IM or constants, A, B and C are natural numbers 1~9, 0.
3. When M ≦ m, A and B are the same values; when M ≧ m, B and C are the same values. A, B and C are written in.
4. A, B and C can be the same as the ST # for GOTO.
5. When calculations are performed without a jump destination, the program calculation is stopped.

[I] and [IM]

1. When the memory number of the memory to be used is put into the I memory, IM can be used instead of that number.
2. The I memory stores the natural numbers 1~9, 0. For other values only the first digit is stored (1 if 10).
3. When the calculator reads ENT I: , the ENT-lamp and "E" light up on the 1st read-out part (this is not an error).

[SUB #]

1. When SUB # N: is placed at the beginning of a program and that program is put at the end of a main program, it is called a subroutine program.
2. When it is desired to execute a subroutine program in a main program, assemble GOTO N: . The N number is the same as the N of SUB # N: . A conditional jump ([IF]) can also be made to a subroutine.

3. Up to 10 subroutines can be assembled but the same number as the N in SUB # N: cannot be used.
(Consequently, [ST #] and [SUB #] together will total 10.)
4. When a subroutine program has been executed (go to the next SUB # N: or read to the 127th step), an automatic return is made to the step after the one from which the main program jumped. (This is not ST # N: .)
5. The destination of [GOTO] and [IF] jumps in a subroutine are in that subroutine. The destination of a [GOTO] of [IF] jump by a main routine cannot be in a subroutine.
6. A subroutine cannot be called by another subroutine.

Operation of the program switch at COMPute

1. When performing automatic calculations (program calculations) using a program stored in the calculator, the program switch must be set at COMPute [STA] is pressed; otherwise, calculations will not be performed.
2. For program calculations either the (ENT) or the (ANS) lamp will light.
3. When [AC] is pressed in program calculation mode, the program preparation mode will first be entered and the [ENT] and [ANS] keys will not operate. (The (ENT) and (ANS) lamps also go out.)
4. The program preparation mode is also entered when a program is executed breaking the programming rules.
5. Manual arithmetic and function calculations are possible in either program mode or program preparation mode but the data memories and independent memory cannot be used.
(To see the contents of the data memories, set to **MANUAL** . Because [MAC] clears all the data memories in **COMPute**, both the data and the answer are erased when it is pressed during a calculation.)
6. When an answer is displayed in the program calculation mode ((ANS) lamp lit), an entry is made and a program executed, the content of the memory number changes to the entered value.

Part 3

APPLICATIONS PACKAGE

Mathematical Programs

3.1 SOLUTION OF TWO SIMULTANEOUS EQUATIONS FOR TWO UNKNOWNS

Given the coefficients in the two simultaneous equations for x_1 and x_2 given by

$$a_{11}x_1 + a_{12}x_2 = b_1$$
$$a_{21}x_1 + a_{22}x_2 = b_2$$

the values of x_1 and x_2 are computed, using

$$x_1 = \frac{a_{22}b_1 - a_{12}b_2}{D}$$

$$x_2 = \frac{b_1 - a_{11}x_1}{a_{12}}$$

where $D = a_{11}a_{22} - a_{12}a_{21}$

D is the determinant of the matrix of coefficients. If D = 0 the computation halts and E appears in the display.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter value of a_{11}	ENT
2	ENT	2	Enter value of a_{12}	ENT
3	ENT	3	Enter value of a_{21}	ENT
4	ENT	4	Enter value of a_{22}	ENT
5	ENT	5	Enter value of b_1	ENT
6	ENT	6	Enter value of b_2	ENT
7	ANS	8	Value of x_1 displayed	ANS
8	ANS	9	Value of x_2 displayed	ANS
9	ANS	0	Value of D displayed.	
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		=	87		:
2		1	45		K	88		9
3		:	46		1	89		:
4		2	47		÷	90		0
5		:	48		K	91		:
6		3	49		0	92		
7		:	50		:	93		
8		4	51		ST #	94		
9		:	52		2	95		
10		5	53		:	96		
11		:	54		8	97		
12		6	55		=	98		
13		:	56		4	99		
14		0	57		X	100		
15		=	58		5	101		
16		2	59		:	102		
17		X	60		9	103		
18		3	61		=	104		
19		:	62		2	105		
20		0	63		X	106		
21		=	64		6	107		
22		1	65		:	108		
23		X	66		8	109		
24		4	67		=	110		
25		—	68		8	111		
26		0	69		—	112		
27		:	70		9	113		
28		IF	71		÷	114		
29		0	72		0	115		
30		=	73		:	116		
31		K	74		9	117		
32		0	75		=	118		
33		:	76		1	119		
34		2	77		+/—	120		
35		:	78		X	121		
36		1	79		8	122		
37		:	80		+	123		
38		2	81		5	124		
39		:	82		÷	125		
40		ST #	83		2	126		
41		1	84		:	127		
42		:	85		ANS	128		
43		0	86		8	129		

3.2 SOLUTION OF A QUADRATIC EQUATION

This program computes the roots of the quadratic equation

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

with $a \neq 0$.

If $b^2 \geqslant 4ac$, so that the roots are real, the roots are calculated from

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

and displayed.

If $b^2 < 4ac$, so that the roots are complex the roots can be written as

$$x = \frac{-b}{2a} \pm i \sqrt{\left(\frac{c}{a} - \left(\frac{b}{2a}\right)^2\right)}$$

and in this case the real part is displayed first, followed by the imaginary part.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter value of a	ENT
2	ENT	2	Enter value of b	ENT
3	ENT	3	Enter value of c	ENT
4	ANS	8	If roots real, displays value of $x_1 = (-b + \sqrt{\ \ }\ )/2a$	ANS
5	ANS	9	If roots are real, displays value of x_2	
6	ANS	5	If the roots are complex, the real part of the roots is displayed	ANS
7	ANS	7	If the roots are complex the imaginary part is displayed	
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		:	87		
2		ENT	45		ST #	88		
3		1	46		1	89		
4		:	47		:	90		
5		2	48		7	91		
6		:	49		=	92		
7		3	50		6	93		
8		:	51		+/−	94		
9		4	52		√	95		
10		=	53		:	96		
11		3	54		GOTO	97		
12		÷	55		3	98		
13		1	56		:	99		
14		:	57		ST #	100		
15		5	58		2	101		
16		=	59		:	102		
17		2	60		8	103		
18		÷	61		=	104		
19		1	62		5	105		
20		÷	63		+	106		
21		K	64		6	107		
22		2	65		√	108		
23		+/−	66		:	109		
24		:	67		9	110		
25		6	68		=	111		
26		=	69		5	112		
27		5	70		−	113		
28		X	71		6	114		
29		5	72		√	115		
30		−	73		:	116		
31		4	74		ANS	117		
32		:	75		8	118		
33		IF	76		:	119		
34		6	77		9	120		
35		=	78		:	121		
36		K	79		ST #	122		
37		.0	80		3	123		
38		:	81		:	124		
39		1	82		ANS	125		
40		:	83		5	126		
41		2	84		:	127		
42		:	85		7	128		
43		2	86		:	129		

3.3 GREATEST COMMON DIVISOR AND LEAST COMMON MULTIPLE

This program computes the Greatest Common Divisor of two positive integers m and n using Euclid's Algorithm. It then finds the Least Common Multiple of m and n by dividing their product by their G.C.D.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter m	ENT
2	ENT	1	Enter n	ENT
3	ANS	1	G.C.D. of m and n	ANS
4	ANS	3	L.C.M. of m and n	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		:	87		1
2		0	45		ST #	88		=
3		:	46		2	89		2
4		1	47		:	90		:
5		:	48		2	91		GOTO
6		3	49		=	92		1
7		=	50		2	93		:
8		0	51		−	94		ST #
9		X	52		1	95		5
10		1	53		:	96		:
11		:	54		IF	97		3
12		IF	55		2	98		=
13		0	56		=	99		3
14		=	57		1	100		÷
15		1	58		:	101		1
16		:	59		3	102		:
17		6	60		:	103		ANS
18		:	61		5	104		1
19		1	62		:	105		:
20		:	63		2	106		3
21		1	64		:	107		:
22		:	65		ST #	108		
23		ST #	66		3	109		
24		6	67		:	110		
25		:	68		IF	111		
26		4	69		2	112		
27		=	70		=	113		
28		1	71		K	114		
29		:	72		0	115		
30		1	73		:	116		
31		=	74		4	117		
32		0	75		:	118		
33		:	76		5	119		
34		0	77		:	120		
35		=	78		4	121		
36		4	79		:	122		
37		:	80		ST #	123		
38		ST #	81		4	124		
39		1	82		:	125		
40		:	83		0	126		
41		2	84		=	127		
42		=	85		1	128		
43		0	86		:	129		

3.4 COMPLEX ARITHMETIC

This program computes the product and quotient of two complex numbers, and displays successively the real and imaginary parts of the answers. The computations are as follows:

$$(a + jb) \times (c + jd) = (ac - bd) + j\,(ad + bc)$$

$$(a + jb) \div (c + jd) = \frac{ac + bd}{c^2 + d^2} + j\,\frac{(bc - ad)}{c^2 + d^2}$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter value of a	ENT
2	ENT	2	Enter value of b	ENT
3	ENT	3	Enter value of c	ENT
4	ENT	4	Enter value of d	ENT
5	ANS	1	Displays real part of product	ANS
6	ANS	2	Displays imaginary part of product	ANS
7	ANS	3	Displays real part of quotient	ANS
8	ANS	4	Displays imaginary part of quotient	
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		x^y	87		
2		1	45		K	88		
3		:	46		2	89		
4		2	47		+	90		
5		:	48		9	91		
6		3	49		:	92		
7		:	50		1	93		
8		4	51		=	94		
9		:	52		0	95		
10		0	53		−	96		
11		=	54		6	97		
12		1	55		:	98		
13		X	56		2	99		
14		3	57		=	100		
15		:	58		5	101		
16		5	59		+	102		
17		=	60		7	103		
18		1	61		:	104		
19		X	62		3	105		
20		4	63		=	106		
21		:	64		0	107		
22		6	65		+	108		
23		=	66		6	109		
24		2	67		÷	110		
25		X	68		9	111		
26		4	69		:	112		
27		:	70		4	113		
28		7	71		=	114		
29		=	72		7	115		
30		2	73		−	116		
31		X	74		5	117		
32		3	75		÷	118		
33		:	76		9	119		
34		9	77		:	120		
35		=	78		ANS	121		
36		4	79		1	122		
37		x^y	80		:	123		
38		K	81		2	124		
39		2	82		:	125		
40		:	83		3	126		
41		9	84		:	127		
42		=	85		4	128		
43		3	86		:	129		

3.5 MODULUS AND ARGUMENT OF A COMPLEX NUMBER

Given the values of R and X, this program computes the modulus, r, and the argument, θ, of the complex number

$$Z = R + jX.$$

The program finds r and θ, where

$$Z = re^{j\theta}$$

using the formulae

$$r = \sqrt{X^2 + R^2}$$

$$\theta = \tan^{-1}\left(\frac{X}{R}\right)$$

The program may also be used to compute the magnitude and angle of an impedance, Z, given by

$$Z = R + jX.$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of R	ENT
2	ENT	2	Enter the value of X	ENT
3	ANS	3	Displays the value of r	ANS
4	ANS	4	Displays the value of θ	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44			87		
2		1	45			88		
3		:	46			89		
4		2	47			90		
5		:	48			91		
6		3	49			92		
7		=	50			93		
8		1	51			94		
9		X	52			95		
10		1	53			96		
11		:	54			97		
12		3	55			98		
13		=	56			99		
14		2	57			100		
15		X	58			101		
16		2	59			102		
17		+	60			103		
18		3	61			104		
19		:	62			105		
20		3	63			106		
21		=	64			107		
22		3	65			108		
23		√	66			109		
24		:	67			110		
25		0	68			111		
26		=	69			112		
27		2	70			113		
28		÷	71			114		
29		1	72			115		
30		:	73			116		
31		4	74			117		
32		=	75			118		
33		0	76			119		
34		arc	77			120		
35		tan	78			121		
36		:	79			122		
37		ANS	80			123		
38		3	81			124		
39		:	82			125		
40		4	83			126		
41		:	84			127		
42			85			128		
43			86			129		

3.6 LOGARITHMS TO ANY BASE

Given the values of m and X, this program computes the value of $\log_m X$ using the formula

$$\log_m X = \frac{\log_{10} X}{\log_{10} m}$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of X	ENT
2	ENT	1	Enter the value of m	ENT
3	ANS	2	Displays the value of $\log_m X$	
4				
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44			87		
2		0	45			88		
3		:	46			89		
4		1	47			90		
5		:	48			91		
6		2	49			92		
7		=	50			93		
8		0	51			94		
9		log	52			95		
10		:	53			96		
11		3	54			97		
12		=	55			98		
13		1	56			99		
14		log	57			100		
15		:	58			101		
16		2	59			102		
17		=	60			103		
18		2	61			104		
19		÷	62			105		
20		3	63			106		
21		:	64			107		
22		ANS	65			108		
23		2	66			109		
24		:	67			110		
25			68			111		
26			69			112		
27			70			113		
28			71			114		
29			72			115		
30			73			116		
31			74			117		
32			75			118		
33			76			119		
34			77			120		
35			78			121		
36			79			122		
37			80			123		
38			81			124		
39			82			125		
40			83			126		
41			84			127		
42			85			128		
43			86			129		

3.7 HYPERBOLIC FUNCTIONS

This program computes the hyperbolic functions using

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

$$\cosh x = \frac{e^x + e^{-x}}{2}$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

To compute sinh x, put 1 in the I-memory, to compute cosh x, put 2 in the I-memory and to compute tanh x, put 3 in the I-memory.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter 1 for sinh x, 2 for cosh x or 3 for tanh x	ENT
2	ENT	4	Enter the value of x	ENT
3	ANS	5	Displays the value of sinh x, cosh x or tanh x	
4				
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		K	87		
2		ENT	45		2	88		
3		I	46		:	89		
4		:	47		GOTO	90		
5		4	48		4	91		
6		:	49		:	92		
7		5	50		ST #	93		
8		=	51		2	94		
9		4	52		:	95		
10		e^x	53		5	96		
11		−	54		=	97		
12		4	55		6	98		
13		+/−	56		÷	99		
14		e^x	57		K	100		
15		:	58		2	101		
16		6	59		:	102		
17		=	60		GOTO	103		
18		4	61		4	104		
19		e^x	62		:	105		
20		+	63		ST #	106		
21		4	64		3	107		
22		+/−	65		:	108		
23		e^x	66		5	109		
24		:	67		=	110		
25		IF	68		5	111		
26		I	69		÷	112		
27		=	70		6	113		
28		K	71		:	114		
29		2	72		ST #	115		
30		:	73		4	116		
31		1	74		:	117		
32		:	75		ANS	118		
33		2	76		5	119		
34		:	77		:	120		
35		3	78			121		
36		:	79			122		
37		ST #	80			123		
38		1	81			124		
39		:	82			125		
40		5	83			126		
41		=	84			127		
42		5	85			128		
43		÷	86			129		

3.8 INVERSE HYPERBOLIC FUNCTIONS

This program computes the values of the inverse hyperbolic functions from

$$\sinh^{-1} x = \ln (x + \sqrt{x^2 + 1})$$

$$\cosh^{-1} x = \ln (x + \sqrt{x^2 - 1})$$

$$\tanh^{-1} x = \frac{1}{2} \ln \left(\frac{1 + x}{1 - x} \right) \text{ for } x < 1$$

Putting 1 in the I-memory causes the program to compute $\sinh^{-1} x$, while 2 in the I-memory causes the computation of $\cosh^{-1} x$ and 3 in the I-memory leads to the computation of $\tanh^{-1} x$.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter 1 for $\sinh^{-1} x$, 2 for $\cosh^{-1} x$, and 3 for $\tanh^{-1} x$	ENT
2	ENT	4	Enter value of x	ENT
3	ANS	5	Displays value of $\sinh^{-1} x$, $\cosh^{-1} x$ or $\tanh^{-1} x$	
4				
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		1	87		÷
2		ENT	45		:	88		K
3		I	46		ST #	89		2
4		:	47		4	90		:
5		4	48		:	91		ST #
6		:	49		5	92		5
7		IF	50		=	93		:
8		I	51		5	94		ANS
9		=	52		√	95		5
10		K	53		+	96		:
11		2	54		4	97		
12		:	55		:	98		
13		1	56		5	99		
14		:	57		=	100		
15		2	58		5	101		
16		:	59		ln	102		
17		3	60		:	103		
18		:	61		GOTO	104		
19		ST #	62		5	105		
20		1	63		:	106		
21		:	64		ST #	107		
22		5	65		3	108		
23		-	66		:	109		
24		4	67		5	110		
25		X	68		=	111		
26		4	69		K	112		
27		+	70		1	113		
28		K	71		−	114		
29		1	72		4	115		
30		:	73		:	116		
31		GOTO	74		5	117		
32		4	75		=	118		
33		:	76		K	119		
34		ST #	77		1	120		
35		2	78		+	121		
36		:	79		4	122		
37		5	80		÷	123		
38		=	81		5	124		
39		4	82		:	125		
40		X	83		5	126		
41		4	84		=	127		
42		−	85		5	128		
43		K	86		ln	129		

3.9 SIMPSON'S RULE

This program uses Simpson's Rule to compute an approximation to the value of a definite integral.

A definite integral is approximated by

$$\int_a^b f(x)\,dx \triangleq \frac{h}{3}\left[f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n\right]$$

Where n must be even

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

$x_0 = a$

$x_n = b$

$x_i = a + ih$ for $i = 0 \ldots n$

$f_i = f(x_i)$ for $i = 0 \ldots n$

The function is entered as subroutine 1, and the program as listed is for $f(x) = x^2 + x$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter value of a	ENT
2	ENT	1	Enter value of b	ENT
3	ENT	2	Enter value of n	ENT
4	ANS	4	Displays the value of the integral	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		2	87		ST #
2		0	45		=	88		4
3		:	46		2	89		:
4		1	47		−	90		0
5		:	48		K	91		=
6		2	49		2	92		1
7		:	50		:	93		:
8		3	51		IF	94		GOTO
9		=	52		2	95		1
10		1	53		=	96		:
11		−	54		K	97		4
12		0	55		0	98		=
13		÷	56		:	99		4
14		2	57		3	100		+
15		:	58		:	101		9
16		GOTO	59		4	102		X
17		1	60		:	103		3
18		:	61		3	104		÷
19		4	62		:	105		K
20		=	63		ST #	106		3
21		9	64		3	107		:
22		:	65		:	108		ANS
23		ST #	66		0	109		4
24		2	67		=	110		:
25		:	68		0	111		SUB #
26		0	69		+	112		1
27		=	70		3	113		:
28		0	71		:	114		9
29		+	72		GOTO	115		=
30		3	73		1	116		0
31		:	74		:	117		X
32		GOTO	75		4	118		0
33		1	76		=	119		+
34		:	77		9	120		0
35		4	78		X	121		:
36		=	79		K	122		
37		9	80		2	123		
38		X	81		+	124		
39		K	82		4	125		
40		4	83		:	126		
41		+	84		GOTO	127		
42		4	85		2	128		
43		:	86		:	129		

3.10 THE NUMERICAL SOLUTION OF A FIRST ORDER DIFFERENTIAL EQUATION

This program computes the solution of an initial value problem of the form $y' = f(x, y)$ with initial condition $y(x_0) = y_0$.

The solution is computed step by step using a third order Runge-Kutta Method given by $y_{n+1} = y_n + \frac{1}{6}k_1 + \frac{2}{3}k_2 + \frac{1}{6}k_3$ for $x_n = x_0 + nh$, $n = 1, 2, \ldots$

where $k_1 = h\,f(x_n, y_n)$

$$k_2 = h\,f(x_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}k_1)$$

$$k_3 = h\,f(x_n + h,\ y_n - k_1 + 2k_2)$$

and h is an increment in x to be specified.

The program as listed will compute an approximation to the solution of $y' = x + y$. However, any other function $f(x, y)$ may be entered by inserting $hf(x, y)$ following SUB # 1: in steps 104 to 127.

Once started from the initial condition the program can be made to compute successively y_n for $n = 1, 2, 3 \ldots$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter value of x_0	ENT
2	ENT	1	Enter value of y_0	ENT
3	ENT	2	Enter value of h	ENT
4	ANS	0	Displays $x_0 + h = x_1$	ANS
5	ANS	1	Displays y_1	ANS
6	ANS	0	Displays $x_n = x_0 + nh$	ANS
7	ANS	1	Displays y_n for $n = 2, 3, \ldots$	ANS
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		GOTO	87		2
2		0	45		1	88		+
3		:	46		:	89		7
4		1	47		4	90		+
5		:	48		=	91		6
6		2	49		9	92		:
7		:	50		:	93		ANS
8		8	51		0	94		0
9		=	52		=	95		:
10		2	53		0	96		1
11		÷	54		+	97		:
12		K	55		8	98		GOTO
13		2	56		:	99		2
14		:	57		1	100		:
15		ST #	58		=	101		SUB #
16		2	59		4	102		1
17		:	60		X	103		:
18		6	61		K	104		9
19		=	62		2	105		=
20		1	63		−	106		0
21		:	64		3	107		+
22		GOTO	65		+	108		1
23		1	66		6	109		X
24		:	67		:	110		2
25		3	68		GOTO	111		:
26		=	69		1	112		
27		9	70		:	113		
28		:	71		7	114		
29		0	72		=	115		
30		=	73		3	116		
31		0	74		+	117		
32		+	75		4	118		
33		8	76		+	119		
34		:	77		9	120		
35		1	78		÷	121		
36		=	79		K	122		
37		3	80		6	123		
38		÷	81		:	124		
39		K	82		1	125		
40		2	83		=	126		
41		+	84		4	127		
42		6	85		÷	128		
43		:	86		K	129		

3.11 THE SOLUTION OF a cos θ – b sin θ = c

Given the values of a, b and c this program computes two adjacent values of θ, in degrees, that satisfy the equation

$$a \cos\theta - b \sin\theta = c.$$

The program computes the value of r and ϕ such that

$$a \cos\theta - b \sin\theta = r \cos(\theta + \phi)$$

from

$$r = \sqrt{a^2 + b^2}$$

$$\phi = \tan^{-1}\left(\frac{b}{a}\right).$$

The program then computes two adjacent roots of the equation using

$$\theta_1 = \cos^{-1}\left(\frac{c}{r}\right) - \phi$$

$$\theta_2 = 360 - \cos^{-1}\left(\frac{c}{r}\right) - \phi.$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of a	ENT
2	ENT	1	Enter the value of b	ENT
3	ENT	2	Enter the value of c	ENT
4	ANS	6	Displays the value of θ (in degrees)	ANS
5	ANS	7	Displays the value of an adjacent root in degrees.	
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		:	87		
2		0	45		5	88		
3		:	46		=	89		
4		1	47		5	90		
5		:	48		ARC	91		
6		2	49		COS	92		
7		:	50		:	93		
8		4	51		6	94		
9		=	52		=	95		
10		1	53		5	96		
11		÷	54		–	97		
12		0	55		4	98		
13		:	56		:	99		
14		4	57		7	100		
15		=	58		=	101		
16		4	59		K	102		
17		ARC	60		3	103		
18		TAN	61		6	104		
19		:	62		0	105		
20		3	63		–	106		
21		=	64		5	107		
22		0	65		–	108		
23		X	66		4	109		
24		0	67		:	110		
25		:	68		ANS	111		
26		3	69		6	112		
27		=	70		:	113		
28		1	71		7	114		
29		X	72		:	115		
30		1	73			116		
31		+	74			117		
32		3	75			118		
33		:	76			119		
34		3	77			120		
35		=	78			121		
36		3	79			122		
37		√	80			123		
38		:	81			124		
39		5	82			125		
40		=	83			126		
41		2	84			127		
42		÷	85			128		
43		3	86			129		

Statistics Programs

3.12 MEAN AND STANDARD DEVIATION

Given a set of statistical values, x_i for $i = 1 \ldots n$, this program computes their mean, standard deviation and variance using the formulae

$$m = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\sigma_1 = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}{n-1}}$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2 - m^2}$$

Variance $= \sigma^2$.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of x_1	ENT
2	ENT	1	Enter the value of x_2 for $i = 1 \ldots n$	ENT
3			⋮	
4	ENT	1	Enter value of x_n	ENT
5	ENT	1		MJ
6	ANS	1	Displays the value of m	ANS
7	ANS	2	Displays the value of σ_1	ANS
8	ANS	4	Displays the value of σ	ANS
9	ANS	3	Displays the value of σ^2	
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		2	87		3
2		ST #	45		÷	88		:
3		1	46		4	89		
4		:	47		:	90		
5		ENT	48		1	91		
6		1	49		=	92		
7		:	50		2	93		
8		2	51		÷	94		
9		=	52		4	95		
10		1	53		:	96		
11		+	54		2	97		
12		2	55		=	98		
13		:	56		3	99		
14		3	57		—	100		
15		=	58		6	101		
16		1	59		÷	102		
17		X	60		5	103		
18		1	61		:	104		
19		+	62		3	105		
20		3	63		=	106		
21		:	64		3	107		
22		4	65		—	108		
23		=	66		6	109		
24		4	67		÷	110		
25		+	68		4	111		
26		K	69		:	112		
27		1	70		2	113		
28		:	71		=	114		
29		GOTO	72		2	115		
30		1	73		√	116		
31		:	74		:	117		
32		MJ	75		4	118		
33		5	76		=	119		
34		=	77		3	120		
35		4	78		√	121		
36		—	79		:	122		
37		K	80		ANS	123		
38		1	81		1	124		
39		:	82		:	125		
40		6	83		2	126		
41		=	84		:	127		
42		2	85		4	128		
43		X	86		:	129		

3.13 GEOMETRIC MEAN

Given a series of values x_i for $i = 1 \dots n$, this program computes their geometric mean, given by

$$G_m = \sqrt[n]{x_1 . x_2 . x_3 \dots x_n}$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of x_1	ENT
2	ENT	1	Enter the value of x_2	ENT
3			⋮	
4	ENT	1	Enter the value of x_n	ENT
5	ENT	1		MJ
6	ANS	2	Displays the value of G_m	
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		2	87		
2		2	45		:	88		
3		=	46			89		
4		K	47			90		
5		1	48			91		
6		:	49			92		
7		ST #	50			93		
8		1	51			94		
9		:	52			95		
10		ENT	53			96		
11		1	54			97		
12		:	55			98		
13		2	56			99		
14		=	57			100		
15		2	58			101		
16		X	59			102		
17		1	60			103		
18		:	61			104		
19		3	62			105		
20		=	63			106		
21		3	64			107		
22		+	65			108		
23		K	66			109		
24		1	67			110		
25		:	68			111		
26		GOTO	69			112		
27		1	70			113		
28		:	71			114		
29		MJ	72			115		
30		3	73			116		
31		=	74			117		
32		K	75			118		
33		1	76			119		
34		÷	77			120		
35		3	78			121		
36		:	79			122		
37		2	80			123		
38		=	81			124		
39		2	82			125		
40		x^y	83			126		
41		3	84			127		
42		:	85			128		
43		ANS	86			129		

3.14 STATISTICAL NORMALISATION

Given a set of statistical values, x_i for $i = 1 \ldots n$, this program computes their mean, m, and standard deviation, σ. It then normalises the statistical values by creating a new random variable with mean value 50 and standard deviation 10 using the formula

$$X_i = 50 + 10 \frac{(x_i - m)}{\sigma}.$$

Subsequently, given a value, x_i, the program computes the value of the corresponding normalised random variable, X_i.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of x_1	ENT
2	ENT	1	Enter the value of x_2	ENT
3			⋮	
4	ENT	1	Enter value of x_n	ENT
5	ENT	1		MJ
6	ANS	1	Displays the value of m	ANS
7	ANS	2	Displays the value of σ	ANS
8	ENT	9	Enter the value of x_i	ENT
9	ANS	3	Displays the value of X_i	
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		÷	87		:
2		ST #	45		4	88		GOTO
3		1	46		−	89		2
4		:	47		3	90		:
5		ENT	48		:	91		
6		1	49		2	92		
7		:	50		=	93		
8		2	51		5	94		
9		=	52		+/−	95		
10		1	53		√	96		
11		+	54		÷	97		
12		2	55		4	98		
13		:	56		√	99		
14		3	57		:	100		
15		=	58		ANS	101		
16		1	59		1	102		
17		X	60		:	103		
18		1	61		2	104		
19		+	62		:	105		
20		3	63		ST #	106		
21		:	64		2	107		
22		4	65		:	108		
23		=	66		ENT	109		
24		4	67		9	110		
25		+	68		:	111		
26		K	69		3	112		
27		1	70		=	113		
28		:	71		9	114		
29		GOTO	72		−	115		
30		1	73		1	116		
31		:	74		÷	117		
32		MJ	75		2	118		
33		1	76		X	119		
34		=	77		K	120		
35		2	78		1	121		
36		÷	79		0	122		
37		4	80		+	123		
38		:	81		K	124		
39		5	82		5	125		
40		=	83		0	126		
41		2	84		:	127		
42		X	85		ANS	128		
43		2	86		3	129		

3.15 CORRELATION COEFFICIENT

Given two sets of data, x_i and y_i for $i = 1 \ldots n$, this program computes their correlation coefficient, γ, using the formula

$$\gamma = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\left[n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right]\left[n \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} y_i^2\right]}$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of x_1	ENT
2	ENT	2	Enter the value of y_1	ENT
3	ENT	1	Enter the value of x_2	ENT
4	ENT	2	Enter the value of y_2	ENT
5	ENT	1	⋮	
6	ENT	1	Enter value of x_n	ENT
7	ENT	2	Enter value of y_n	ENT
8	ENT	1		MJ
9	ANS	1	Displays the value of γ	
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		7	87		5
2		ST #	45		:	88		−
3		1	46		8	89		9
4		:	47		=	90		X
5		ENT	48		8	91		2
6		1	49		+	92		:
7		:	50		K	93		1
8		2	51		1	94		=
9		:	52		:	95		8
10		3	53		GOTO	96		X
11		=	54		1	97		7
12		3	55		:	98		−
13		+	56		MJ	99		3
14		1	57		9	100		÷
15		:	58		=	101		2
16		4	59		3	102		√
17		=	60		X	103		:
18		4	61		3	104		ANS
19		+	62		:	105		1
20		2	63		0	106		:
21		:	64		=	107		
22		5	65		4	108		
23		=	66		X	109		
24		1	67		4	110		
25		X	68		:	111		
26		1	69		3	112		
27		+	70		=	113		
28		5	71		3	114		
29		:	72		X	115		
30		6	73		4	116		
31		=	74		:	117		
32		2	75		2	118		
33		X	76		=	119		
34		2	77		8	120		
35		+	78		X	121		
36		6	79		6	122		
37		:	80		−	123		
38		7	81		0	124		
39		=	82		:	125		
40		1	83		2	126		
41		X	84		=	127		
42		2	85		8	128		
43		+	86		X	129		

3.16 CLASSIFICATION

Given data classified according to some code, this program computes the total data in each group and also sums the data in all the groups. Thus, given data classified as follows:

Classification Code	Amount of data
1	59
7	73
4	19
...	...

the program will sum the data in each group for up to nine classification groups. Each group must be given an integer from 1 to 9 as its classification code.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter the classification code	ENT
2	ENT	i	For i from 1 to 9 inclusive enter the corresponding data	ENT
3	ENT	E	Continue until all data	ENT
4	ENT	i	is entered	ENT
5	ENT	E		MJ
6	ANS	1	Displays total of group with code 1	ANS
7	ANS	2	Displays total of group with code 2	ANS
8	⋮		⋮	
9	ANS	9	Displays total of group with code 9	ANS
10	ANS	0	Displays the total of all data in all groups.	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		ANS	87		
2		ST #	45		IM	88		
3		1	46		:	89		
4		:	47		0	90		
5		ENT	48		=	91		
6		I	49		0	92		
7		:	50		+	93		
8		0	51		IM	94		
9		=	52		:	95		
10		IM	53		IF	96		
11		:	54		I	97		
12		ENT	55		=	98		
13		IM	56		K	99		
14		:	57		9	100		
15		IM	58		:	101		
16		=	59		2	102		
17		IM	60		:	103		
18		+	61		3	104		
19		0	62		:	105		
20		:	63		3	106		
21		GOTO	64		:	107		
22		1	65		ST #	108		
23		:	66		3	109		
24		MJ	67		:	110		
25		0	68		ANS	111		
26		=	69		0	112		
27		K	70		:	113		
28		0	71			114		
29		:	72			115		
30		I	73			116		
31		=	74			117		
32		0	75			118		
33		:	76			119		
34		ST #	77			120		
35		2	78			121		
36		:	79			122		
37		I	80			123		
38		=	81			124		
39		I	82			125		
40		+	83			126		
41		K	84			127		
42		1	85			128		
43		:	86			129		

3.17 LINEAR REGRESSION

This program computes the least squares fit straight line for a set of n data points given by (x_i, y_i) for $i = 1 \ldots n$. The straight line is denoted by

$$y = m x + c$$

and its parameters are given by

$$m = \frac{\frac{1}{n} \sum_{i=1}^{n} x_i y_i - \bar{x}.\bar{y}}{\frac{1}{n} \sum_{i=1}^{n} x_i^2 - \bar{x}^2}$$

$$c = \bar{y} - m \bar{x} \qquad \text{where } \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \text{ and } \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

When the values of the parameters of the regression line have been computed, values of x can be entered and the corresponding values of y given by the regression line are computed. This program can also be used for trend line analysis.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of x_1	ENT
2	ENT	2	Enter the value of y_1	ENT
3	ENT	1	Enter the value of x_i	ENT
4	ENT	2	Enter the value of y_i	ENT
5	ENT	1	Enter the value of x_n	ENT
6	ENT	2	Enter the value of y_n	ENT
7	ENT	1		MJ
8	ANS	0	Displays the value of m	ANS
9	ANS	1	Displays the value of c	ANS
10	ENT	9	Enter the value of x	ENT
11	ANS	2	Displays the value of y predicted by the regression.	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		7	87		2
2		ST #	45		:	88		=
3		1	46		8	89		3
4		:	47		=	90		X
5		ENT	48		K	91		7
6		1	49		1	92		:
7		:	50		+	93		1
8		2	51		8	94		=
9		:	52		:	95		5
10		3	53		GOTO	96		X
11		=	54		1	97		4
12		1	55		:	98		—
13		+	56		MJ	99		2
14		3	57		9	100		÷
15		:	58		=	101		9
16		4	59		3	102		:
17		=	60		X	103		ANS
18		2	61		3	104		0
19		+	62		:	105		:
20		4	63		9	106		1
21		:	64		=	107		:
22		5	65		8	108		ST #
23		=	66		X	109		2
24		1	67		5	110		:
25		X	68		—	111		ENT
26		1	69		9	112		9
27		+	70		:	113		:
28		5	71		0	114		2
29		:	72		=	115		=
30		6	73		3	116		9
31		=	74		X	117		X
32		2	75		4	118		0
33		X	76		:	119		+
34		2	77		0	120		1
35		+	78		=	121		:
36		6	79		8	122		ANS
37		:	80		X	123		2
38		7	81		7	124		:
39		=	82		—	125		GOTO
40		1	83		0	126		2
41		X	84		÷	127		:
42		2	85		9	128		
43		+	86		:	129		

3.18 CURVE FITTING

This program computes the least squares fit curve of the form

$$y = a x^b$$

for a set of data points given by (x_i, y_i) for $i = 1 \ldots n$.

The formulae used are

$$b = \frac{\sum_{i=1}^{n} (\ln x_i)(\ln y_i) - \frac{1}{n}\left(\sum_{i=1}^{n} \ln x_i\right)\left(\sum_{i=1}^{n} \ln y_i\right)}{\sum_{i=1}^{n} (\ln x_i)^2 - \frac{1}{n}\left(\sum_{i=1}^{n} \ln x_i\right)^2}$$

$$a = \exp\left[\frac{1}{n}\sum_{i=1}^{n} \ln y_i - \frac{b}{n}\sum_{i=1}^{n} \ln x_i\right]$$

Subsequently, given values of x the program computes the corresponding value of y on the fitted curve.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of x_1	ENT
2	ENT	2	Enter the value of y_1	ENT
3	ENT	1	Enter the value of x_2	ENT
4	ENT	2	Enter the value of y_2	ENT
5	ENT	1	⋮	
6	ENT	1	Enter value of x_n	ENT
7	ENT	2	Enter value of y_n	ENT
8	ENT	1		MJ
9	ANS	9	Displays the value of a	ANS
10	ANS	0	Displays the value of b	ANS
11	ENT	1	Enter a value of x	ENT
12	ANS	2	Displays the corresponding value of y on the curve.	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		:	87		9
2		ST #	45		8	88		=
3		1	46		=	89		3
4		:	47		4	90		÷
5		ENT	48		X	91		6
6		1	49		4	92		−
7		:	50		+	93		9
8		2	51		8	94		:
9		:	52		:	95		9
10		3	53		GOTO	96		=
11		=	54		1	97		9
12		2	55		:	98		e^x
13		ln	56		MJ	99		:
14		+	57		0	100		ANS
15		3	58		=	101		9
16		:	59		5	102		:
17		4	60		X	103		0
18		=	61		5	104		:
19		1	62		÷	105		ST #
20		ln	63		6	106		2
21		:	64		−	107		:
22		5	65		8	108		ENT
23		=	66		:	109		1
24		4	67		0	110		:
25		+	68		=	111		2
26		5	69		5	112		=
27		:	70		X	113		1
28		6	71		3	114		x^y
29		=	72		÷	115		0
30		6	73		6	116		X
31		+	74		−	117		9
32		K	75		7	118		:
33		1	76		÷	119		ANS
34		:	77		0	120		2
35		7	78		:	121		:
36		=	79		9	122		GOTO
37		1	80		=	123		2
38		ln	81		5	124		:
39		X	82		÷	125		
40		2	83		6	126		
41		ln	84		X	127		
42		+	85		0	128		
43		7	86		:	129		

3.19 COMBINATIONS AND PERMUTATIONS

This program includes a subroutine to compute factorials. It calls this subroutine in order to compute the number of combinations of n objects taken r at a time

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

and the number of permutations of n objects taken r at a time

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

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of n	ENT
2	ENT	2	Enter the value of r	ENT
3	ANS	6	Displays the value of nCr	ANS
4	ANS	7	Displays the value of nPr	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		6	87		2
2		ENT	45		=	88		:
3		1	46		7	89		1
4		:	47		÷	90		:
5		2	48		5	91		ST #
6		:	49		:	92		2
7		8	50		ANS	93		:
8		=	51		6	94		
9		1	52		:	95		
10		:	53		7	96		
11		GOTO	54		:	97		
12		0	55		SUB #	98		
13		:	56		0	99		
14		4	57		:	100		
15		=	58		9	101		
16		9	59		=	102		
17		:	60		K	103		
18		8	61		1	104		
19		=	62		:	105		
20		2	63		ST #	106		
21		:	64		1	107		
22		GOTO	65		:	108		
23		0	66		9	109		
24		:	67		=	110		
25		5	68		9	111		
26		=	69		X	112		
27		9	70		8	113		
28		:	71		:	114		
29		8	72		8	115		
30		=	73		=	116		
31		1	74		8	117		
32		−	75		−	118		
33		2	76		K	119		
34		:	77		1	120		
35		GOTO	78		:	121		
36		0	79		IF	122		
37		:	80		8	123		
38		7	81		=	124		
39		=	82		K	125		
40		4	83		1	126		
41		÷	84		:	127		
42		9	85		2	128		
43		:	86		:	129		

3.20 PERMUTATIONS

Given n and r this program computes $_nP_r$, the number of permutations of objects taken r at a time from a group of size n. This number is given by

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

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of n	
2	ENT	2	Enter the value of r	
3	ANS	4	Displays the value of nP r.	
4				
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		:	87		
2		ENT	45		2	88		
3		1	46		:	89		
4		:	47		ST #	90		
5		2	48		2	91		
6		:	49		:	92		
7		3	50		ANS	93		
8		=	51		4	94		
9		1	52		:	95		
10		−	53			96		
11		2	54			97		
12		+	55			98		
13		K	56			99		
14		1	57			100		
15		:	58			101		
16		4	59			102		
17		=	60			103		
18		1	61			104		
19		:	62			105		
20		ST #	63			106		
21		1	64			107		
22		:	65			108		
23		1	66			109		
24		=	67			110		
25		1	68			111		
26		−	69			112		
27		K	70			113		
28		1	71			114		
29		:	72			115		
30		4	73			116		
31		≠	74			117		
32		4	75			118		
33		X	76			119		
34		1	77			120		
35		:	78			121		
36		IF	79			122		
37		3	80			123		
38		=	81			124		
39		1	82			125		
40		:	83			126		
41		1	84			127		
42		:	85			128		
43		2	86			129		

3.21 COMBINATIONS

Given n and r this program computes ${}_nC_r$, the number of combinations of objects taken r at a time from a group of size n. This number is given by

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

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of n	ENT
2	ENT	2	Enter the value of r	ENT
3	ANS	6	Displays the value of ${}_nC_r$	
4				
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		9	87		:
2		ENT	45		:	88		
3		1	46		ANS	89		
4		:	47		6	90		
5		2	48		:	91		
6		:	49		SUB #	92		
7		8	50		0	93		
8		=	51		:	94		
9		1	52		9	95		
10		:	53		=	96		
11		GOTO	54		K	97		
12		0	55		1	98		
13		:	56		:	99		
14		4	57		ST #	100		
15		=	58		1	101		
16		9	59		:	102		
17		:	60		9	103		
18		8	61		=	104		
19		=	62		9	105		
20		2	63		X	106		
21		:	64		8	107		
22		GOTO	65		:	108		
23		0	66		8	109		
24		:	67		=	110		
25		5	68		8	111		
26		=	69		–	112		
27		9	70		K	113		
28		:	71		1	114		
29		8	72		:	115		
30		=	73		IF	116		
31		1	74		8	117		
32		–	75		=	118		
33		2	76		K	119		
34		:	77		1	120		
35		GOTO	78		:	121		
36		0	79		2	122		
37		:	80		:	123		
38		6	81		2	124		
39		=	82		:	125		
40		4	83		1	126		
41		÷	84		:	127		
42		5	85		ST #	128		
43		÷	86		2	129		

3.22 POISSON DISTRIBUTION

This program computes the probability density function, f(x), and the cumulative distribution function, P(x), of the Poisson distribution using the formulae

$$f(x) = \frac{m^x e^{-m}}{x!}$$

$$P(x) = \sum_{k=0}^{x} f(k)$$

where $m = np$

n is the sample size

p is the probability of occurrence of an event

$x = 0, 1, 2 \ldots$

When n is large, greater than 100, say, f(x) gives the probability of x successes in n trials when p is the probability of success at a single trial and the trials are independent. In the same circumstances, P(x) gives the probability of x or fewer successes.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of p	ENT
2	ENT	1	Enter the value of n	ENT
3	ENT	2	Enter the value of x	ENT
4	ANS	4	Displays the value of f(x)	ANS
5	ANS	5	Displays the value of P(x)	
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		ST #	87		
2		ENT	45		2	88		
3		0	46		:	89		
4		:	47		6	90		
5		1	48		=	91		
6		:	49		6	92		
7		2	50		+	93		
8		:	51		K	94		
9		3	52		1	95		
10		=	53		:	96		
11		0	54		4	97		
12		X	55		=	98		
13		1	56		4	99		
14		:	57		X	100		
15		4	58		3	101		
16		=	59		÷	102		
17		3	60		6	103		
18		±	61		:	104		
19		e^x	62		5	105		
20		:	63		=	106		
21		5	64		5	107		
22		=	65		+	108		
23		4	66		4	109		
24		:	67		:	110		
25		6	68		GOTO	111		
26		=	69		1	112		
27		K	70		:	113		
28		0	71		ST #	114		
29		:	72		3	115		
30		ST #	73		:	116		
31		1	74		ANS	117		
32		:	75		4	118		
33		IF	76		:	119		
34		6	77		5	120		
35		=	78		:	121		
36		2	79			122		
37		:	80			123		
38		2	81			124		
39		:	82			125		
40		3	83			126		
41		:	84			127		
42		3	85			128		
43		:	86			129		

3.23 BINOMIAL DISTRIBUTION

This program computes the probability density function, f(x), and the cumulative distribution function, P (n), of the binomial distribution using the formulas.

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x} \qquad \text{for } 0 \leqslant x \leqslant n$$

$$= 0 \quad \text{otherwise}$$

$$P(x) = \sum_{i=0}^{n} f(x) \qquad \text{for } 0 \leqslant x \leqslant n$$

$$= 1 \quad \text{if } x > n,$$

where n is the sample size
p is the probability of occurrence of an event
$x = 0, 1, 2 \ldots$

The probability of x successes in n trials when p is the probability of success at a single trial and the trials are independent is given by f(x). In the same circumstances, P (x) gives the probability of x or fewer successes.

Note that if the sample size n is greater than about a hundred the Poisson distribution program should be used.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of p	ENT
2	ENT	1	Enter the value of n	ENT
3	ENT	2	Enter the value of x ($\geqslant 0$)	ENT
4	ANS	4	Displays the value of f(x)	ANS
5	ANS	5	Displays the value of P(x)	
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		=	87		÷
2		ENT	45		3	88		6
3		0	46		x^y	89		X
4		:	47		1	90		0
5		1	48		:	91		÷
6		:	49		5	92		3
7		2	50		=	93		:
8		:	51		4	94		5
9		IF	52		:	95		=
10		2	53		6	96		6
11		=	54		=	97		+
12		1	55		K	98		4
13		:	56		0	99		:
14		2	57		:	100		1
15		:	58		ST #	101		=
16		2	59		3	102		1
17		:	60		:	103		—
18		1	61		IF	104		K
19		:	62		6	105		1
20		ST #	63		=	106		:
21		1	64		2	107		GOTO
22		:	65		:	108		3
23		5	66		4	109		:
24		=	67		:	110		ST #
25		K	68		5	111		5
26		1	69		:	112		:
27		:	70		5	113		ANS
28		ANS	71		:	114		4
29		4	72		ST #	115		:
30		:	73		4	116		5
31		5	74		:	117		:
32		:	75		6	118		
33		ST #	76		=	119		
34		2	77		6	120		
35		:	78		+	121		
36		3	79		K	122		
37		=	80		1	123		
38		K	81		:	124		
39		1	82		4	125		
40		—	83		=	126		
41		0	84		4	127		
42		:	85		X	128		
43		4	86		1	129		

3.24 NORMAL DISTRIBUTION

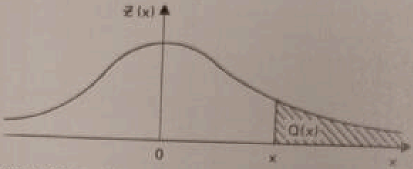
This program computes the values of the probability density function, $Z(x)$, and the cumulative distribution function, $Q(x)$, of the standard normal distribution using the formulae

$$Z(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

$$Q(x) = Z(x)\,[b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5]$$

Where $t = \dfrac{1}{1 + px}$

and
$p = 0.231642$
$b_1 = 0.319382$
$b_2 = -0.356564$
$b_3 = 1.78148$
$b_4 = -1.82126$
$b_5 = 1.33027$



The value of π is entered from the keyboard to memory 1 to maintain accuracy in the computation. The term in the square brackets in the formula for $Q(x)$ is programmed as

$$((((b_5 t + b_4)\,t + b_3)\,t + b_2)\,t + b_1)\,t$$

to take advantage of the expression evaluation method of the calculator.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of x	ENT
2	ENT	1	Enter π	ENT
3	ANS	9	Displays the value of $Z(x)$	ANS
4	ANS	3	Displays the value of $Q(x)$	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

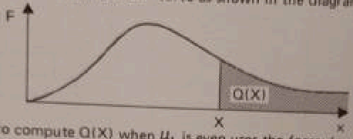
STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		K	87		X
2		0	45		1	88		2
3		:	46		:	89		+
4		1	47		2	90		K
5		:	48		=	91		.
6		9	49		K	92		3
7		=	50		1	93		5
8		0	51		÷	94		6
9		X	52		2	95		5
10		0	53		:	96		6
11		÷	54		3	97		4
12		K	55		=	98		±
13		2	56		K	99		X
14		:	57		1	100		2
15		1	58		.	101		+
16		=	59		3	102		K
17		1	60		3	103		.
18		X	61		0	104		3
19		K	62		2	105		1
20		2	63		7	106		9
21		:	64		X	107		3
22		9	65		2	108		8
23		=	66		+	109		2
24		9	67		K	110		X
25		±	68		1	111		2
26		e^x	69		.	112		X
27		÷	70		8	113		9
28		1	71		2	114		:
29		√	72		1	115		ANS
30		:	73		2	116		9
31		2	74		6	117		:
32		=	75		±	118		3
33		K	76		X	119		:
34		.	77		2	120		
35		2	78		+	121		
36		3	79		K	122		
37		1	80		1	123		
38		6	81		.	124		
39		4	82		7	125		
40		2	83		8	126		
41		X	84		1	127		
42		0	85		4	128		
43		+	86		8	129		

3.25 F DISTRIBUTION

If y and z are independent random variables with chi-square distributions and μ_1 and μ_2 degrees of freedom respectively, then $F = \dfrac{y/\mu_1}{z/\mu_2}$ is called the variance ratio.

Its distribution is the F distribution with μ_1 and μ_2 degrees of freedom.

These programs compute Q(X), the probability that x > X, so that Q(X) is the "upper tail" of the area under the F curve as shown in the diagram.



The program to compute Q(X) when μ_1 is even uses the formula

$$Q(X) = K^{\frac{\mu_2}{2}}\left[1 + \frac{\mu_2(1-K)}{2} + \frac{\mu_2(\mu_2+2)}{2.4}(1-K)^2 + \ldots \right.$$

$$\left. \ldots + \frac{\mu_2(\mu_2+2)\ldots(\mu_2+\mu_1-4)}{2.4\ldots(\mu_1-2)}(1-K)^{\frac{\mu_1-2}{2}}\right]$$

and the program to compute Q(X) when M_2 is even uses the formula

$$Q(X) = 1-(1-K)^{\frac{\mu_1}{2}}\left[1 + \frac{\mu_1 K}{2} + \frac{\mu_1(\mu_1+2)}{2.4}K^2 + \ldots \right.$$

$$\left. \ldots + \frac{\mu_1(\mu_1+2)\ldots(\mu_1+\mu_2-4)}{2.4\ldots(\mu_2-2)}K^{\frac{\mu_2-2}{2}}\right]$$

where $K = \dfrac{1}{1+\mu_1 X/\mu_2}$

1) μ_2 even.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of X	ENT
2	ENT	1	Enter the value of μ_1	ENT
3	ENT	2	Enter the value of μ_2 (even)	ENT
4	ANS	9	Displays the value of Q(X)	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		6	87		X
2		ENT	45		=	88		3
3		0	46		K	89		X
4		:	47		1	90		1
5		1	48		:	91		÷
6		:	49		9	92		8
7		2	50		=	93		:
8		:	51		6	94		9
9		3	52		:	95		=
10		=	53		7	96		9
11		0	54		=	97		+
12		X	55		2	98		6
13		1	56		−	99		:
14		÷	57		K	100		1
15		2	58		2	101		−
16		+	59		:	102		1
17		K	60		ST #	103		+
18		1	61		3	104		K
19		÷	62		:	105		2
20		÷	63		IF	106		:
21		K	64		8	107		GOTO
22		1	65		=	108		3
23		:	66		7	109		:
24		4	67		:	110		ST #
25		=	68		1	111		2
26		K	69		:	112		:
27		1	70		2	113		9
28		−	71		:	114		=
29		3	72		2	115		9
30		:	73		:	116		X
31		5	74		ST #	117		4
32		=	75		1	118		−
33		1	76		:	119		−
34		÷	77		8	120		K
35		K	78		=	121		1
36		2	79		8	122		:
37		:	80		+	123		ANS
38		4	81		K	124		9
39		=	82		2	125		:
40		4	83		:	126		
41		$x^{1/y}$	84		6	127		
42		5	85		=	128		
43		:	86		6	129		

2) μ_1 even.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of X	ENT
2	ENT	1	Enter the value of μ_1 (even)	ENT
3	ENT	2	Enter the value of μ_2	ENT
4	ANS	9	Displays the value of Q(X)	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		6	87		X
2		ENT	45		=	88		4
3		0	46		K	89		X
4		:	47		1	90		2
5		1	48		:	91		÷
6		:	49		9	92		8
7		2	50		=	93		:
8		:	51		6	94		9
9		3	52		:	95		=
10		=	53		7	96		9
11		0	54		=	97		+
12		X	55		1	98		6
13		1	56		—	99		:
14		÷	57		K	100		2
15		2	58		2	101		=
16		+	59		:	102		2
17		K	60		ST #	103		+
18		1	61		3	104		K
19		÷	62		:	105		2
20		÷	63		IF	106		:
21		K	64		8	107		GOTO
22		1	65		=	108		3
23		:	66		7	109		:
24		4	67		:	110		ST #
25		=	68		1	111		2
26		K	69		:	112		:
27		1	70		2	113		9
28		—	71		:	114		=
29		3	72		2	115		9
30		:	73		:	116		X
31		5	74		ST #	117		3
32		=	75		1	118		:
33		2	76		:	119		ANS
34		÷	77		8	120		9
35		K	78		=	121		:
36		2	79		8	122		
37		:	80		+	123		
38		3	81		K	124		
39		=	82		2	125		
40		3	83		:	126		
41		x^y	84		6	127		
42		5	85		=	128		
43		:	86		6	129		

3.26 CHI-SQUARE DISTRIBUTION

This program computes the value of the Chi-square probability density function, f(x), given by the formula

$$f(x) = \frac{\left(\frac{x}{2}\right)^{\frac{r}{2}} e^{\frac{-x}{2}}}{x\,\Gamma\left(\frac{r}{2}\right)}$$

where $x > 0$

r is the number of degrees of freedom ($r \geqslant 2$)

$\Gamma\left(\frac{r}{2}\right) = \left(\frac{r}{2} - 1\right)!$ when r is even

$\Gamma\left(\frac{r}{2}\right) = \left(\frac{r}{2} - 1\right)\left(\frac{r}{2} - 2\right) \cdots \left(\frac{1}{2}\right)\sqrt{\pi}$ when r is odd.

If r is even 0 must be entered in the I-memory. If r is odd, 1 must be entered in the I-memory.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of x	ENT
2	ENT	1	Enter the value of r	ENT
3	ENT	E	Enter 0 if r is even, or 1 if r is odd	ENT
4	ANS	4	Displays the value of f(x)	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		9	87		0
2		0	45		:	88		:
3		:	46		IF	89		9
4		1	47		I	90		=
5		:	48		=	91		K
6		I	49		K	92		1
7		:	50		1	93		:
8		2	51		:	94		ST #
9		=	52		3	95		1
10		1	53		:	96		:
11		÷	54		4	97		9
12		K	55		:	98		=
13		2	56		4	99		0
14		:	57		:	100		X
15		3	58		ST #	101		5
16		=	59		3	102		:
17		0	60		:	103		6
18		÷	61		ANS	104		=
19		K	62		4	105		5
20		2	63		:	106		−
21		:	64		ST #	107		K
22		5	65		4	108		1
23		=	66		:	109		:
24		2	67		4	110		IF
25		−	68		=	111		5
26		K	69		4	112		=
27		1	70		X	113		K
28		:	71		K	114		1
29		GOTO	72		1	115		:
30		0	73		.	116		2
31		:	74		1	117		:
32		4	75		2	118		2
33		=	76		8	119		:
34		3	77		3	120		1
35		x^y	78		7	121		:
36		2	79		9	122		ST #
37		X	80		1	123		2
38		3	81		7	124		:
39		±	82		:	125		
40		e^x	83		ANS	126		
41		÷	84		4	127		
42		0	85		:	128		
43		÷	86		SUB #	129		

3.27 RANDOM NUMBER GENERATOR

This program generates random numbers that are evenly distributed on the interval (0, 1). It uses the formula

$$X_{n+1} = (a X_n + c) \bmod m$$

where X_n is the seed

$a = 24298$

$c = 9991$

$m = 199017.$

This sequence has period m.

Given a seed, this program displays the random number $\frac{X_{n+1}}{m}$, which is in the range (0, 1).

It is possible to use a pair of uniform random numbers on the interval (0, 1), denoted by r_1 and r_2, to generate a normal deviate with zero mean and unit standard deviation using $x = (-2 \ln u_1)^{1/2} \cos(2\pi u_2)$

A relatively small value should be given to the seed, X_n, otherwise the program can take a long time to produce its result. A value of approximately unity is suggested.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter seed, X_n	ENT
2	ANS	1	Displays random number	
3				
4				
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		2	87		
2		0	45		:	88		
3		:	46		ST #	89		
4		1	47		2	90		
5		=	48		:	91		
6		K	49		1	92		
7		2	50		=	93		
8		4	51		1	94		
9		2	52		−	95		
10		9	53		2	96		
11		8	54		:	97		
12		X	55		GOTO	98		
13		0	56		1	99		
14		+	57		:	100		
15		K	58		ST #	101		
16		9	59		3	102		
17		9	60		:	103		
18		9	61		1	104		
19		9	62		=	105		
20		1	63		1	106		
21		:	64		÷	107		
22		2	65		2	108		
23		=	66		:	109		
24		K	67		ANS	110		
25		1	68		1	111		
26		9	69		:	112		
27		9	70			113		
28		0	71			114		
29		1	72			115		
30		7	73			116		
31		:	74			117		
32		ST #	75			118		
33		1	76			119		
34		:	77			120		
35		IF	78			121		
36		1	79			122		
37		=	80			123		
38		2	81			124		
39		:	82			125		
40		3	83			126		
41		:	84			127		
42		2	85			128		
43		:	86			129		

Financial Programs

3.28 COMPOUND INTEREST

Given the values of any three of the four variables involved in compound interest calculations, this program computes the value of the remaining variable. The formulae used are

$$F = P(1 + i)^n \qquad P = \frac{F}{(1 + i)^n}$$

$$n = \frac{\log(F/P)}{\log(1 + i)} \qquad i = 100\left[\left(\frac{F}{P}\right)^{\frac{1}{n}} - 1\right]$$

where

F is the future value
P is the present value
n is the number of periods
i is the interest rate per period entered as a percentage

When 0 is entered in the I-memory, F is computed;
when 1 is entered in the I-memory, P is computed;
when 2 is entered in the I-memory, i is computed and
when 3 is entered in the I-memory, n is computed.

In each case four further data entries are required. The three known values should be inserted in the appropriate memories and zero should be entered as the value of the variable to be computed. The program then computes and displays the value of the unknown variable.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter 0 to compute F, 1 for P, 2 for i or 3 for n	ENT *
2	ENT	0	Enter the value of F	ENT *
3	ENT	1	Enter the value of P	ENT *
4	ENT	2	Enter the value of i	ENT *
5	ENT	3	Enter the value of n	ENT *
6	ANS	0, 1 2 or 3	If 0 displays the value of F, if 1 then P, if 2 then i, if 3 then n	

*When P is being computed, 1 is in the I-memory. When ENT 1 appears enter 0 as the value. Proceed similarly for the other computations.

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		X	87		:
2		I	45		1	88		3
3		:	46		:	89		=
4		0	47		ST #	90		3
5		:	48		2	91		log
6		1	49		:	92		÷
7		:	50		1	93		4
8		2	51		=	94		:
9		:	52		6	95		ST #
10		3	53		x^y	96		4
11		:	54		3	97		:
12		6	55		÷	98		4
13		=	56		÷	99		=
14		2	57		0	100		K
15		÷	58		:	101		1
16		K	59		ST #	102		÷
17		1	60		3	103		3
18		0	61		:	104		:
19		0	62		IF	105		2
20		+	63		I	106		=
21		K	64		=	107		0
22		1	65		K	108		÷
23		:	66		2	109		1
24		IF	67		:	110		x^y
25		I	68		6	111		4
26		=	69		:	112		—
27		K	70		4	113		K
28		1	71		:	114		1
29		:	72		5	115		X
30		1	73		:	116		K
31		:	74		ST #	117		1
32		2	75		5	118		0
33		:	76		:	119		0
34		3	77		4	120		:
35		:	78		=	121		ST #
36		ST #	79		6	122		6
37		1	80		log	123		:
38		:	81		:	124		ANS
39		0	82		3	125		IM
40		=	83		=	126		:
41		6	84		0	127		
42		x^y	85		÷	128		
43		3	86		1	129		

3.29 LOAN REPAYMENT

This program performs computations on loan repayment using the formula

$$P = Q \times \frac{R\,(1+R)^N}{(1+R)^N - 1}$$

$$N = n \times 12$$

$$R = r \div 12$$

where Q is the amount of the loan,
P is the monthly payment,
n is the number of years taken to repay the loan
r is the yearly interest rate, entered as a percentage.

When 9 is initially entered in the I-memory, the program will compute the value of Q given the values of P, n and r. When 8 is placed in the I-memory the value of P will be computed given the values of Q, n and r.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter 8 to compute P, Enter 9 to compute Q	ENT
2	ENT	0	Enter the value of r	ENT
3	ENT	1	Enter the value of n	ENT
4	ENT	2	If I = 9, enter the value of P	ENT
5	ANS	2	Displays the value of Q	
6	ENT	4	If I = 8, enter the value of Q	ENT
7	ANS	4	Displays the value of P	
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		3	87		:
2		I	45		:	88		
3		:	46		IF	89		
4		0	47		I	90		
5		:	48		=	91		
6		1	49		K	92		
7		:	50		8	93		
8		0	51		:	94		
9		=	52		1	95		
10		0	53		:	96		
11		÷	54		2	97		
12		K	55		:	98		
13		1	56		1	99		
14		2	57		:	100		
15		0	58		ST#	101		
16		0	59		2	102		
17		:	60		:	103		
18		1	61		ENT	104		
19		=	62		2	105		
20		1	63		:	106		
21		×	64		2	107		
22		K	65		=	108		
23		1	66		2	109		
24		2	67		×	110		
25		:	68		3	111		
26		3	69		:	112		
27		=	70		ANS	113		
28		K	71		2	114		
29		1	72		:	115		
30		+	73		ST #	116		
31		0	74		1	117		
32		x^y	75		:	118		
33		1	76		ENT	119		
34		:	77		4	120		
35		3	78		:	121		
36		=	79		4	122		
37		3	80		=	123		
38		−	81		4	124		
39		K	82		÷	125		
40		1	83		3	126		
41		÷	84		:	127		
42		0	85		ANS	128		
43		÷	86		4	129		

3.30 ORDINARY ANNUITY OF KNOWN INTEREST RATE

This program can be used when the annuity interest rate is known. It uses the formulae

$$P = pp\,\frac{(1-(1+i)^{-n})}{i} \quad \text{for } i>0$$

$$pp = P\,\frac{i}{1-(1+i)^{-n}} \quad \text{for } i>0$$

$$n = \frac{-\log\left(1-\frac{iP}{pp}\right)}{\log(1+i)} \quad \text{for } i>0$$

where P is the present value
pp is the payment per period
n is the number of periods
i is the interest rate per period, entered as a percentage.

Thus, given i and any two of P, pp and n, the program computes the value of the third.
Entering 1 in the I-memory causes the computation of P,
entering 2 in the I-memory causes the computation of pp,
and entering 3 in the I-memory causes the computation of n.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter 1 to compute P, 2 to compute pp, 3 to compute n	ENT
2	ENT	0	Enter i as a percentage	ENT
3	ENT	1	Enter the value of P	ENT *
4	ENT	2	Enter the value of pp	ENT *
5	ENT	3	Enter the value of n	ENT *
6	ANS	1 2 3	Displays the value of one of { P, pp, n }	
7				
8				
9				
10				

*Enter two of these as the program directs

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		8	87		÷
2		I	45		x^y	88		0
3		:	46		3	89		:
4		0	47		±	90		ANS
5		:	48		:	91		IM
6		0	49		1	92		:
7		=	50		=	93		ST #
8		0	51		K	94		3
9		÷	52		1	95		:
10		K	53		–	96		ENT
11		1	54		9	97		1
12		0	55		X	98		:
13		0	56		2	99		2
14		:	57		÷	100		:
15		8	58		0	101		5
16		=	59		:	102		=
17		0	60		ANS	103		0
18		+	61		IM	104		X
19		K	62		:	105		1
20		1	63		ST #	106		÷
21		:	64		2	107		2
22		IF	65		:	108		–
23		I	66		ENT	109		K
24		=	67		1	110		1
25		K	68		:	111		:
26		2	69		3	112		3
27		:	70		:	113		=
28		1	71		9	114		5
29		:	72		=	115		±
30		2	73		8	116		log
31		:	74		x^y	117		÷
32		3	75		3	118		8
33		:	76		±	119		log
34		ST #	77		:	120		±
35		1	78		2	121		:
36		:	79		=	122		ANS
37		ENT	80		K	123		IM
38		2	81		1	124		:
39		:	82		–	125		
40		3	83		9	126		
41		:	84		÷	127		
42		9	85		1	128		
43		=	86		÷	129		

3.31 ACCRUED INTEREST

This program computes the simple interest accruing on an amount of money using the formula

$$I = \frac{P \times n \times i}{100 \times DPY}$$

where I is the interest accrued over a period
n is the number of days in the period
P is the amount of the principal
i is the annual interest rate expressed as a percentage
DPY is the number of days per year that the interest rate is based on.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of P	ENT
2	ENT	1	Enter the value of n	ENT
3	ENT	2	Enter the value of i (%)	ENT
4	ENT	3	Enter the value of DPY	ENT
5	ANS	4	Displays the value of I	
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44			87		
2		0	45			88		
3		:	46			89		
4		1	47			90		
5		:	48			91		
6		2	49			92		
7		:	50			93		
8		3	51			94		
9		:	52			95		
10		4	53			96		
11		=	54			97		
12		0	55			98		
13		X	56			99		
14		1	57			100		
15		X	58			101		
16		2	59			102		
17		÷	60			103		
18		3	61			104		
19		÷	62			105		
20		K	63			106		
21		1	64			107		
22		0	65			108		
23		0	66			109		
24		:	67			110		
25		ANS	68			111		
26		4	69			112		
27		:	70			113		
28			71			114		
29			72			115		
30			73			116		
31			74			117		
32			75			118		
33			76			119		
34			77			120		
35			78			121		
36			79			122		
37			80			123		
38			81			124		
39			82			125		
40			83			126		
41			84			127		
42			85			128		
43			86			129		

3.32 AMORTISED LOAN SCHEDULE

This program computes an amortised loan schedule. It uses the formulae

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

$$Balance_0 = P$$

and for $j = 1 \ldots n,$

$$I_j = Balance_{j-1} \times i$$

$$Prin_j = PMT - I_j$$

$$I = \sum_{i=1}^{j} I_i$$

$$Balance_j = Balance_{j-1} - Prin_j$$

where P is the principal
i is the interest rate per period (%)
n is the number of periods
PMT is the payment per period
$Balance_j$ is the balance after j payments ($j = 0 \ldots n$)
I_j is the interest accrued in the jth period
$Prin_j$ is the payment to principal in the jth period
I is the interest accruing over the first j periods

The program displays j and $Balance_j$ for $j = 1 \ldots n$. $Balance_n$ is zero to within the accuracy permitted by rounding error. At each stage the other variables are computed. I_j is stored in memory 5, $Prin_j$ is stored in memory 6 and I is in memory 7. They can be displayed by examining the memories in manual mode or by modifying the program.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of P	ENT
2	ENT	1	Enter the value of i (%)	ENT
3	ENT	2	Enter the value of n	ENT
4	ANS	8	Displays 1 (for first payment)	ANS
5	ANS	4	Displays the balance after 1 payment	ANS
6	ANS	8	Displays j for j = 2 . . n	ANS
7	ANS	4	Displays the balance after j payments for j = 1 . . n	ANS
8	ANS	8	Displays n	ANS
9	ANS	4	Displays the zero balance to within rounding error	
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		3	87		:
2		ENT	45		±	88		IF
3		0	46		X	89		8
4		:	47		0	90		=
5		1	48		:	91		2
6		:	49		ST #	92		:
7		2	50		1	93		1
8		:	51		:	94		:
9		8	52		5	95		2
10		=	53		=	96		:
11		K	54		4	97		2
12		0	55		X	98		:
13		:	56		1	99		ST #
14		1	57		:	100		2
15		=	58		6	101		:
16		1	59		=	102		ANS
17		÷	60		3	103		8
18		K	61		—	104		:
19		1	62		5	105		4
20		0	63		:	106		:
21		0	64		7	107		
22		:	65		=	108		
23		3	66		7	109		
24		=	67		+	110		
25		1	68		5	111		
26		+	69		:	112		
27		K	70		4	113		
28		1	71		=	114		
29		:	72		4	115		
30		3	73		—	116		
31		=	74		6	117		
32		3	75		:	118		
33		x^y	76		8	119		
34		2	77		=	120		
35		±	78		8	121		
36		—	79		+	122		
37		K	80		K	123		
38		1	81		1	124		
39		:	82		:	125		
40		3	83		ANS	126		
41		=	84		8	127		
42		1	85		:	128		
43		÷	86		4	129		

3.33 SINKING FUND OF KNOWN INTEREST RATE

A desired future value can be accumulated with fixed payments at regular intervals and a fixed interest rate. This is known as a sinking fund. Given the values of the interest rate and of two of the other variables, this program computes the value of the remaining variable. The formulae used are

$$F = PMT \times \left(\frac{(1+i)^n - 1}{i}\right) \qquad \text{for } i > 0$$

$$PMT = \frac{F \times i}{(1+i)^n - 1} \qquad \text{for } i > 0$$

$$n = \frac{\log\left[\frac{F \times i}{PMT} + 1\right]}{\log(i+1)} \qquad \text{for } i > 0$$

where F is the future value
PMT is the payment per period
i is the interest rate per period, expressed as a %
n is the number of payments

When 1 is entered in the I-memory the program computes the value of F, when 2 is entered in the I-memory the value of PMT is computed and entering 3 in the I-memory leads to the computation of n.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY	
				START	
1	ENT	E	Enter 1 to compute F, 2 to compute PMT, 3 to compute n	ENT	
2	ENT	0	Enter the value of i (%)	ENT	
3	ENT	1	Enter the value of F	ENT	*
4	ENT	2	Enter the value of PMT	ENT	*
5	ENT	3	Enter the value of n	ENT	*
6	ANS	1 2 3	Displays the value of F / PMT / n		
7					
8					

*Enter two of these as the program directs.

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		8	87		3
2		I	45		x^y	88		:
3		:	46		3	89		ENT
4		0	47		—	90		1
5		:	48		K	91		:
6		0	49		1	92		2
7		=	50		÷	93		:
8		0	51		0	94		3
9		÷	52		X	95		=
10		K	53		2	96		1
11		1	54		:	97		X
12		0	55		ANS	98		0
13		0	56		IM	99		÷
14		:	57		:	100		2
15		8	58		ST #	101		+
16		=	59		2	102		K
17		K	60		:	103		1
18		1	61		ENT	104		:
19		+	62		1	105		3
20		0	63		:	106		=
21		:	64		3	107		3
22		IF	65		:	108		log
23		I	66		2	109		÷
24		=	67		=	110		8
25		K	68		8	111		log
26		2	69		x^y	112		:
27		:	70		3	113		ANS
28		1	71		—	114		IM
29		:	72		K	115		:
30		2	73		1	116		
31		:	74		:	117		
32		3	75		2	118		
33		:	76		=	119		
34		ST #	77		1	120		
35		1	78		X	121		
36		:	79		0	122		
37		ENT	80		÷	123		
38		2	81		2	124		
39		:	82		:	125		
40		3	83		ANS	126		
41		:	84		IM	127		
42		1	85		:	128		
43		=	86		ST #	129		

3.34 SINKING FUND INTEREST DETERMINATION

This program computes the interest rate per period for a sinking fund given the values of:

F — the future value

PMT — the payment per period

n — the number of payments

The interest rate is computed iteratively using the Newton-Raphson method. The formulae used are

$$i_{k+1} = i_k - \frac{f(i_k)}{f'(i_k)} \qquad \text{for } k = 0, 1, 2 \ldots$$

where i is the interest rate, displayed as a percentage

i_k is the kth approximation to i for k = 0, 1 . . .

$$f(i) = \frac{PMT\,[\,(1+i)^n - 1\,]}{i} - F$$

$$i_o = \frac{F}{PMT\,.\,n^2} - \frac{PMT}{F}$$ is the initial approximation.

The program displays the initial approximation and then displays successive approximations. When two successive approximations agree to the required accuracy, they may be taken as the value required for i.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of F	ENT
2	ENT	1	Enter the value of PMT	ENT
3	ENT	2	Enter the value of n	ENT
4	ANS	9	Displays initial approximation to i	ANS
5	ANS	9	Displays improved approximation to i	ANS
6	ANS	9	— " —	
7				
8				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		K	87		÷
2		0	45		1	88		3
3		:	46		0	89		−
4		1	47		0	90		7
5		:	48		:	91		:
6		2	49		ANS	92		3
7		:	50		9	93		=
8		3	51		:	94		5
9		=	52		4	95		−
10		1	53		=	96		0
11		÷	54		K	97		÷
12		0	55		1	98		7
13		:	56		+	99		+
14		4	57		3	100		3
15		=	58		:	101		:
16		0	59		5	102		GOTO
17		÷	60		=	103		1
18		1	61		4	104		:
19		÷	62		x^y	105		
20		2	63		2	106		
21		÷	64		−	107		
22		2	65		K	108		
23		:	66		1	109		
24		3	67		X	110		
25		=	68		1	111		
26		4	69		÷	112		
27		−	70		3	113		
28		3	71		:	114		
29		:	72		7	115		
30		6	73		=	116		
31		=	74		4	117		
32		2	75		x^y	118		
33		−	76		6	119		
34		K	77		X	120		
35		1	78		2	121		
36		:	79		X	122		
37		ST #	80		1	123		
38		1	81		÷	124		
39		:	82		3	125		
40		9	83		:	126		
41		=	84		7	127		
42		3	85		=	128		
43		X	86		5	129		

3.35 SINKING FUND – ALTERNATIVE

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of F	ENT
2	ENT	1	Enter the value of PMT	ENT
3	ENT	2	Enter the value of n	ENT
4	ENT	8	Enter the tolerance on i as a fraction of 1	ENT
5	ANS	9	Displays the value of i obtained to the required accuracy	
6				
7				
8				
9				
10				

PROGRAM LISTING

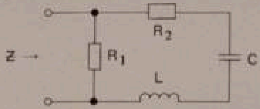
STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		ST #	87		3
2		0	45		1	88		=
3		:	46		:	89		5
4		1	47		4	90		−
5		:	48		=	91		0
6		2	49		K	92		÷
7		:	50		1	93		7
8		8	51		+	94		+
9		:	52		3	95		3
10		3	53		:	96		:
11		=	54		5	97		7
12		1	55		=	98		=
13		÷	56		4	99		7
14		0	57		x^y	100		cos
15		:	58		2	101		:
16		4	59		−	102		IF
17		=	60		K	103		7
18		0	61		1	104		=
19		÷	62		×	105		8
20		1	63		1	106		:
21		÷	64		÷	107		1
22		2	65		3	108		:
23		÷	66		:	109		2
24		2	67		7	110		:
25		:	68		=	111		2
26		3	69		4	112		:
27		=	70		x^y	113		ST #
28		4	71		6	114		2
29		−	72		×	115		:
30		3	73		2	116		9
31		:	74		×	117		=
32		6	75		1	118		3
33		=	76		÷	119		×
34		2	77		3	120		K
35		−	78		:	121		1
36		K	79		7	122		0
37		1	80		=	123		0
38		:	81		5	124		:
39		8	82		÷	125		ANS
40		=	83		3	126		9
41		8	84		−	127		:
42		cos	85		7	128		
43		:	86		:	129		

N.B. This program uses cos – the only even function on the machine to terminate the iterations. The problem is that an error usually occurs because the argument of the cos is often out of range.

Electrical Engineering Programs

3.36 SERIES RESONANT CIRCUIT

Given the circuit shown



this program will compute its resonant frequency, f_s, and its input impedance, Z, at a given frequency, f. The formulae used are

$$F_s = \frac{1}{2\pi\sqrt{LC}}$$

and

$$\frac{1}{Z} = \frac{1}{R_1} + \frac{1}{R_2 + j\left(\omega L - \frac{1}{\omega C}\right)} = \frac{1}{|Z|} e^{-j\theta_z}$$

where $\omega = 2\pi f$

f is the input frequency (Hz)

L is the inductor value (henrys)

C is the capacitor value (farads)

R_1, R_2 are resistor values (ohms)

$|Z|$ is the magnitude of the impedance at f

θ_z is the phase angle of the impedance at f

The value of 2π is inserted in memory 9 initially for accuracy of computation.

USER INSTRUCTIONS

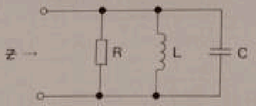
STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of L	ENT
2	ENT	1	Enter the value of C	ENT
3	ENT	2	Enter the value of R_1	ENT
4	ENT	3	Enter the value of R_2	ENT
5	ENT	9	Enter 2π	ENT
6	ANS	4	Displays the value of fs	ANS
7	ENT	5	Enter the value of f	ENT
8	ANS	4	Displays the value of $\|Z\|$	ANS
9	ANS	6	Displays the value of θ_z	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		1	87		7
2		0	45		:	88		:
3		:	46		6	89		4
4		1	47		=	90		=
5		:	48		5	91		7
6		2	49		X	92		X
7		:	50		0	93		7
8		3	51		—	94		:
9		:	52		6	95		4
10		9	53		:	96		=
11		:	54		7	97		8
12		4	55		=	98		X
13		=	56		6	99		8
14		K	57		X	100		+
15		1	58		6	101		4
16		÷	59		:	102		:
17		9	60		7	103		4
18		÷	61		=	104		=
19		0	62		3	105		K
20		√	63		X	106		1
21		÷	64		3	107		÷
22		1	65		+	108		4
23		√	66		7	109		√
24		:	67		:	110		:
25		ANS	68		8	111		6
26		4	69		=	112		=
27		:	70		6	113		8
28		ENT	71		÷	114		÷
29		5	72		7	115		7
30		:	73		:	116		:
31		5	74		7	117		6
32		=	75		=	118		=
33		9	76		3	119		6
34		X	77		÷	120		ARC
35		5	78		7	121		TAN
36		:	79		:	122		:
37		6	80		7	123		ANS
38		=	81		=	124		4
39		K	82		K	125		:
40		1	83		1	126		6
41		÷	84		÷	127		:
42		5	85		2	128		
43		÷	86		+	129		

3.37 PARALLEL RESONANT CIRCUIT

For the following circuit



this program computes the resonant frequency, f_R, and the impedance, Z, at a given frequency, f. The program uses the formulae

$$f_R = \frac{1}{2\pi\sqrt{LC}}$$

and $$\frac{1}{Z} = \frac{1}{R} + j\left(\omega C - \frac{1}{\omega L}\right) = \frac{1}{|Z|}e^{-j\theta_z}$$

where $\omega = 2\pi f$

f is the input frequency (Hz)

R is the resistance (ohms)

C is the capacitance (farads)

L is the inductance (henrys)

$|Z|$ is the magnitude of the impedance

θ_z is the phase angle of the impedance.

The value of 2π is entered in memory 9 initially for accuracy of computation.

USER INSTRUCTIONS

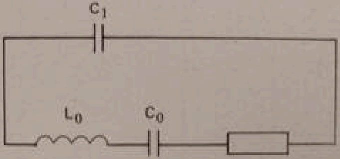
STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of L	ENT
2	ENT	1	Enter the value of C	ENT
3	ENT	2	Enter the value of R	ENT
4	ENT	9	Enter 2π	ENT
5	ANS	3	Displays the value of f_R	ANS
6	ENT	4	Enter the value of f	ENT
7	ANS	7	Displays the value of $\|Z\|$	ANS
8	ANS	8	Displays the value of θ_z	ANS
9	ENT	4	....	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		:	87		÷
2		0	45		6	88		5
3		:	46		=	89		:
4		1	47		4	90		8
5		:	48		X	91		=
6		2	49		1	92		8
7		:	50		:	93		ARC
8		9	51		6	94		TAN
9		:	52		=	95		:
10		3	53		K	96		ANS
11		=	54		1	97		7
12		K	55		÷	98		:
13		1	56		4	99		8
14		÷	57		÷	100		:
15		9	58		0	101		GOTO
16		÷	59		—	102		1
17		0	60		6	103		:
18		√	61		:	104		
19		÷	62		7	105		
20		1	63		=	106		
21		√	64		6	107		
22		:	65		X	108		
23		ANS	66		6	109		
24		3	67		:	110		
25		:	68		7	111		
26		ST #	69		=	112		
27		1	70		5	113		
28		:	71		X	114		
29		ENT	72		5	115		
30		4	73		+	116		
31		:	74		7	117		
32		4	75		:	118		
33		=	76		7	119		
34		4	77		=	120		
35		X	78		K	121		
36		9	79		1	122		
37		:	80		÷	123		
38		5	81		7	124		
39		=	82		√	125		
40		K	83		:	126		
41		1	84		8	127		
42		÷	85		=	128		
43		2	86		6	129		

3.38 FREQUENCY RANGE OF A CRYSTAL OSCILLATOR CIRCUIT

The frequency range of a crystal oscillator circuit, such as the one represented in the diagram, lies between its series resonant frequency and its parallel resonant frequency.



Given the values of

L_o in henrys
C_o in farads, and
C_1 in farads

this program computes the series resonant frequency, f_s, and the parallel resonant frequency, fp, from the formulae

$$fs = \frac{1}{2\pi\sqrt{L_o C_o}} \qquad fp = 2\pi \frac{1}{\sqrt{\dfrac{L_o C_o C_1}{(C_o + C_1)}}}$$

USER INSTRUCTIONS

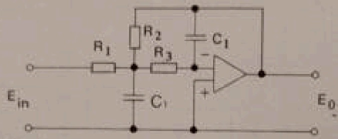
STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of L_o	ENT
2	ENT	2	Enter the value of C_o	ENT
3	ENT	3	Enter the value of C_1	ENT
4	ANS	5	Displays the value of fs	ANS
5	ANS	8	Displays the value of fp	
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		:	87		8
2		1	45		7	88		:
3		:	46		=	89		
4		2	47		1	90		
5		:	48		X	91		
6		3	49		2	92		
7		:	50		X	93		
8		4	51		3	94		
9		=	52		÷	95		
10		1	53		6	96		
11		X	54		:	97		
12		2	55		7	98		
13		:	56		=	99		
14		5	57		7	100		
15		=	58		√	101		
16		K	59		:	102		
17		2	60		8	103		
18		X	61		=	104		
19		K	62		K	105		
20		3	63		2	106		
21		.	64		X	107		
22		1	65		K	108		
23		4	66		3	109		
24		1	67		.	110		
25		5	68		1	111		
26		9	69		4	112		
27		2	70		1	113		
28		X	71		5	114		
29		4	72		9	115		
30		√	73		2	116		
31		:	74		X	117		
32		5	75		7	118		
33		=	76		:	119		
34		K	77		8	120		
35		1	78		=	121		
36		÷	79		K	122		
37		5	80		1	123		
38		:	81		÷	124		
39		6	82		8	125		
40		=	83		:	126		
41		2	84		ANS	127		
42		+	85		5	128		
43		3	86		:	129		

3.39 LOW-PASS ACTIVE FILTER

This program is for implementation of a low-pass filter using the following configuration:



The transfer function is:

$$\frac{E_0\ (s)}{E_{in}\ (s)} = \frac{-1/R_1R_3C_1C_2}{s^2 + \left(\frac{1}{R_1} + \frac{1}{R_3} + \frac{1}{R_2}\right)\left(\frac{1}{C_1}\right) s + \frac{1}{R_2R_3C_1C_2}}$$

which is of the form:

$$H\ (s) = \frac{-H_0\ \omega_0^2}{s^2 + \alpha\omega_0\ s + \omega_0^2}$$

where: H_0 = Pass band gain
$\omega_0 = 2\pi f_0$
f_0 = Cutoff frequency in hertz
α = Peaking factor = 2ζ
ζ = Damping ratio

Given α, H_0, f_0 in hertz, and C_2 in farads the problem calculates $R_1 R_2 R_3$ and C_1 according to the formulae:

$$C_1 = \frac{4\ (1 + H_0)\ C_2}{\alpha^2} \qquad R_2 = \frac{\alpha}{4\pi f_0 C_2}$$

$$R_1 = \frac{R_2}{H_0} \qquad R_3 = \frac{R_2}{H_0 + 1}$$

NOTES:

- All input values must be positive.
- Because operational amplifiers are non-ideal, H_0 should be chosen to be less than 10 when α is about 0.1. H_0 may be increased if α is also increased. A maximum H_0 of 100 is acceptable with α = 1 for an operational amplifier with an open loop pass band gain of at least 80 dB.

USER INSTRUCTIONS

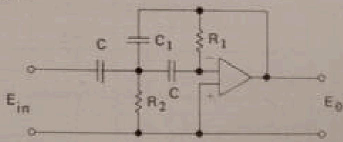
STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of α	ENT
2	ENT	1	Enter the value of H_o	ENT
3	ENT	2	Enter the value of f_o	ENT
4	ENT	3	Enter the value of C_2	ENT
5	ANS	4	Displays the value of C_1	ANS
6	ANS	5	Displays the value of R_1	ANS
7	ANS	6	Displays the value of R_2	ANS
8	ANS	7	Displays the value of R_3	
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		5	87		
2		0	45		=	88		
3		:	46		6	89		
4		1	47		÷	90		
5		:	48		1	91		
6		2	49		:	92		
7		:	50		7	93		
8		3	51		=	94		
9		:	52		K	95		
10		4	53		1	96		
11		=	54		+	97		
12		K	55		1	98		
13		1	56		÷	99		
14		+	57		÷	100		
15		1	58		6	101		
16		X	59		:	102		
17		K	60		ANS	103		
18		4	61		4	104		
19		X	62		:	105		
20		3	63		5	106		
21		÷	64		:	107		
22		0	65		6	108		
23		÷	66		:	109		
24		0	67		7	110		
25		:	68		:	111		
26		6	69			112		
27		=	70			113		
28		0	71			114		
29		÷	72			115		
30		2	73			116		
31		÷	74			117		
32		3	75			118		
33		÷	76			119		
34		K	77			120		
35		1	78			121		
36		2	79			122		
37		.	80			123		
38		5	81			124		
39		6	82			125		
40		6	83			126		
41		3	84			127		
42		7	85			128		
43		:	86			129		

3.40 HIGH-PASS ACTIVE FILTER

This program is for implementation of a high-pass filter using the following configuration:



The transfer function is:

$$\frac{E_0\ (s)}{E_{in}\ (s)} = \frac{-\frac{C}{C_1}s^2}{s^2 + \left(\frac{2}{C_1} + \frac{1}{C}\right)\left(\frac{1}{R_1}\right)s + \frac{1}{R_1 R_2 C C_1}}$$

which is the form:

$$H\ (s) = \frac{-H_0 s^2}{s^2 + \alpha\omega_0\ s + \omega_0^2}$$

where: H_0 = Pass band again; α = Peaking factor = 2ζ

$\omega_0 = 2\pi f_0$; ζ = Damping ratio

f_0 = Cutoff frequency in hertz

Given α, H_0, f_0 in hertz and C in farads the program calculates R_1, R_2, and C_1 according to the formulae:

$$R_1 = \frac{2H_0 + 1}{\alpha 2\pi f_0 C} \qquad R_2 = \frac{\alpha}{2\pi f_0 C\ (2 + 1/H_0)} \qquad C_1 = \frac{C}{H_0}$$

NOTES:

• All input values must be positive. • Because operational amplifiers are non-ideal, H_0 should be chosen to be less than 10 when α is about 0.1. H_0 may be increased if α is also increased. A maximum H_0 of 100 is acceptable with α = 1 for an operational amplifier with an open-loop pass band gain of at least 80 dB. • If the values of the variables are restricted to available components, no problem with overflow will occur.

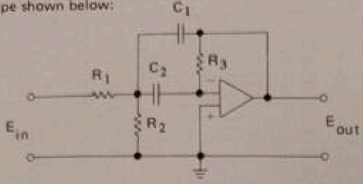
STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of α	ENT
2	ENT	1	Enter the value of H_0	ENT
3	ENT	2	Enter the value of f_0	ENT
4	ENT	3	Enter the value of C	ENT
5	ANS	4	Displays the value of R_1	ANS
6	ANS	5	Displays the value of R_2	ANS
7	ANS	6	Displays the value of C_1	
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		5	87		
2		0	45		=	88		
3		:	46		0	89		
4		1	47		X	90		
5		:	48		1	91		
6		2	49		÷	92		
7		:	50		9	93		
8		3	51		÷	94		
9		:	52		3	95		
10		7	53		÷	96		
11		=	54		7	97		
12		K	55		:	98		
13		2	56		6	99		
14		X	57		=	100		
15		1	58		3	101		
16		+	59		÷	102		
17		K	60		1	103		
18		1	61		:	104		
19		:	62		ANS	105		
20		9	63		4	106		
21		=	64		:	107		
22		K	65		5	108		
23		6	66		:	109		
24		.	67		6	110		
25		2	68		:	111		
26		8	69			112		
27		3	70			113		
28		1	71			114		
29		8	72			115		
30		5	73			116		
31		X	74			117		
32		2	75			118		
33		:	76			119		
34		4	77			120		
35		=	78			121		
36		7	79			122		
37		÷	80			123		
38		0	81			124		
39		÷	82			125		
40		9	83			126		
41		÷	84			127		
42		3	85			128		
43		:	86			129		

3.41 ACTIVE BANDPASS FILTER DESIGN

Given f_o, Q, C_1, C_2, and A_o, this program will calculate R_1, R_2 and R_3 for the circuit type shown below:



where: $\omega_o = 2\pi f_o$ = radian frequency
f_o = centre frequency of passband (hertz)
Q = quality factor
A_o = voltage multiplication factor
C_1, C_2 = capacitance in farads
R_1, R_2, R_3 = resistance in ohms

Given f_o, Q, C_1, C_2, and A_o, this program will calculate R_1, R_2 and R_3 using the formulae:

$$R_1 = \frac{Q}{|A_o|\omega_o C_1} \qquad R_2 = \left(Q(C_1 + C_2)\omega_o - \frac{1}{R_1}\right)^{-1} \qquad R_3 = \frac{Q}{\omega_o}\left(\frac{1}{C_1} + \frac{1}{C_2}\right)$$

NOTES:

- If $Q < \sqrt{\frac{|A_o|}{2}}$, a zero will appear in the display after A_o is entered. Either Q or A_o should be adjusted so that $Q > \sqrt{\frac{|A_o|}{2}}$.

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of f_0	ENT
2	ENT	1	Enter the value of Q	ENT
3	ENT	2	Enter the value of C_1	ENT
4	ENT	3	Enter the value of C_2	ENT
5	ENT	4	Enter the value of $\lvert A_0 \rvert$	ENT*
6	ANS	6	Displays the value of R_1	ANS
7	ANS	7	Displays the value of R_2	ANS
8	ANS	8	Displays the value of R_3	
9				
10	ANS	5	*If $Q < \sqrt{\lvert A_0 \rvert / 2}$ 0 is displayed	

PROGRAM LISTING

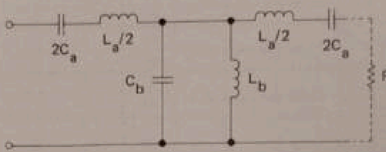
STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		:	87		7
2		0	45		2	88		=
3		:	46		:	89		K
4		1	47		2	90		1
5		:	48		:	91		−
6		2	49		ST #	92		6
7		:	50		1	93		:
8		3	51		:	94		7
9		:	52		5	95		=
10		4	53		=	96		2
11		:	54		K	97		+
12		0	55		0	98		3
13		=	56		:	99		X
14		0	57		ANS	100		1
15		X	58		5	101		X
16		K	59		:	102		0
17		6	60		ST #	103		−
18		.	61		2	104		7
19		2	62		:	105		÷
20		8	63		6	106		÷
21		3	64		=	107		K
22		1	65		1	108		1
23		8	66		÷	109		:
24		5	67		4	110		ANS
25		:	68		÷	111		6
26		5	69		0	112		:
27		=	70		÷	113		7
28		4	71		2	114		:
29		÷	72		:	115		8
30		K	73		8	116		:
31		2	74		=	117		
32		:	75		2	118		
33		5	76		+	119		
34		=	77		3	120		
35		5	78		÷	121		
36		√	79		2	122		
37		:	80		÷	123		
38		IF	81		3	124		
39		1	82		÷	125		
40		=	83		0	126		
41		5	84		X	127		
42		:	85		1	128		
43		1	86		:	129		

3.42 PASSIVE BANDPASS FILTER DESIGN

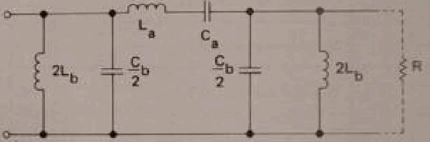
This program and the next relate to the following typical passive bandpass filter circuits.

Typical Circuits

T-section



π-section



The equations used are:

$$C_a = \frac{f_2 - f_1}{4\pi f_1 f_2 R} \qquad C_b = \frac{1}{\pi (f_2 - f_1) R}$$

$$L_a = \frac{R}{\pi (f_2 - f_1)} \qquad L_b = \frac{R(f_2 - f_1)}{4\pi f_1 f_2}$$

where:
- f_1 = low cutoff frequency (Hz) $\qquad f_2 > f_1 > 0$
- f_2 = high cutoff frequency (Hz)
- C = capacitance (farads)
- L = inductance (henrys)
- R = image impedance at mid-frequency (ohms) $\qquad R > 0$

The first program computes the ideal component values for the T-section and π-section filters, given f_1, f_2 and R.
The second program computes f_1, f_2 and R when the component values are known.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of f_1	ENT
2	ENT	1	Enter the value of f_2	ENT
3	ENT	2	Enter the value of R	ENT
4	ENT	5	Enter π	ENT
5	ANS	6	Displays the value of C_a	ANS
6	ANS	7	Displays the value of L_a	ANS
7	ANS	8	Displays the value of C_b	ANS
8	ANS	9	Displays the value of L_b	
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		=	87		
2		0	45		K	88		
3		:	46		1	89		
4		1	47		÷	90		
5		:	48		5	91		
6		2	49		÷	92		
7		:	50		4	93		
8		5	51		÷	94		
9		:	52		2	95		
10		3	53		:	96		
11		=	54		9	97		
12		0	55		=	98		
13		X	56		2	99		
14		1	57		X	100		
15		X	58		4	101		
16		5	59		÷	102		
17		X	60		3	103		
18		K	61		:	104		
19		4	62		ANS	105		
20		:	63		6	106		
21		4	64		:	107		
22		=	65		7	108		
23		1	66		:	109		
24		−	67		8	110		
25		0	68		:	111		
26		:	69		9	112		
27		6	70		:	113		
28		=	71			114		
29		4	72			115		
30		÷	73			116		
31		3	74			117		
32		÷	75			118		
33		2	76			119		
34		:	77			120		
35		7	78			121		
36		=	79			122		
37		2	80			123		
38		÷	81			124		
39		5	82			125		
40		÷	83			126		
41		4	84			127		
42		:	85			128		
43		8	86			129		

3.42 PASSIVE BANDPASS FILTER DESIGN

Program 2

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	6	Enter the value of C_a	ENT
2	ENT	7	Enter the value of L_a	ENT
3	ENT	8	Enter the value of C_b	ENT
4	ENT	9	Enter the value of L_b	ENT
5	ENT	5	Enter π	ENT
6	ANS	0	Displays the value of f_1	ANS
7	ANS	1	Displays the value of f_2	ANS
8	ANS	2	Displays the value of R	
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		÷	87		
2		6	45		K	88		
3		:	46		2	89		
4		7	47		÷	90		
5		:	48		5	91		
6		8	49		:	92		
7		:	50		1	93		
8		9	51		=	94		
9		:	52		4	95		
10		5	53		√	96		
11		:	54		+	97		
12		2	55		3	98		
13		=	56		÷	99		
14		7	57		K	100		
15		√	58		2	101		
16		÷	59		÷	102		
17		8	60		5	103		
18		√	61		:	104		
19		:	62		ANS	105		
20		3	63		0	106		
21		=	64		:	107		
22		2	65		1	108		
23		÷	66		:	109		
24		7	67		2	110		
25		:	68		:	111		
26		4	69			112		
27		=	70			113		
28		7	71			114		
29		+	72			115		
30		9	73			116		
31		÷	74			117		
32		8	75			118		
33		÷	76			119		
34		7	77			120		
35		÷	78			121		
36		9	79			122		
37		:	80			123		
38		0	81			124		
39		=	82			125		
40		4	83			126		
41		√	84			127		
42		−	85			128		
43		3	86			129		

3.43 TRANSMISSION LINE IMPEDANCE

This program can be used to calculate the high frequency characteristic impedance (Z_o) for the following types of transmission lines:

Coaxial line

$$Z_0\ (\text{coax}) = \frac{K}{\sqrt{\epsilon_r}} \log \frac{D}{d}, \text{ where } K = \frac{\sqrt{\mu_0}}{2\pi\sqrt{\epsilon_0}\log e} = 138.0598238$$

D = inner diameter of outer conductor $\qquad \epsilon_0$ = permittivity of free space

d = outer diameter of inner conductor $\qquad \epsilon_r > 0$

ϵ_r = relative permittivity of the dielectric medium $\qquad D \geqslant d > 0$

μ_0 = permeability of free space

Two-conductor line

$$Z_0\ (\text{2-con}) = \frac{2K}{\sqrt{\epsilon_r}} \log \left(\frac{D}{d} + \sqrt{\left(\frac{D}{d}\right)^2 - 1} \right)$$

where D = centre-to-centre conductor spacing

d = conductor diameter

K, ϵ_r as for coaxial line

Single conductor near ground

$$Z_0\ (\text{1-con}) = \frac{K}{\sqrt{\epsilon_r}} \log \left(\frac{4D}{d}\right)$$

where D = spacing of centre of conductor to ground

d = conductor diameter

K, ϵ_r as for coaxial line

When 1 is placed in the I-memory, the program computes Z_0 (coax); when 2 is placed in the I-memory the program computes Z_0 (2-con) and placing 3 in the I-memory causes the computation of Z_0 (1-con).

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter 1 for Z (coax), 2 for Z (2-con), 3 for Z (1-con)	ENT
2	ENT	9	Enter the value of K = 138.059824	ENT
3	ENT	8	Enter the value of D	ENT
4	ENT	7	Enter the value of d	ENT
5	ENT	6	Enter the value of ϵ_r	ENT
6	ANS	1 2 3	Displays the value of Z (coax) Z (2-con) Z (1-con)	

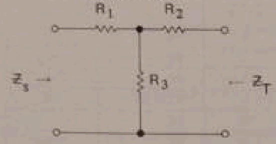
PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		X	87		K
2		I	45		4	88		4
3		:	46		:	89		X
4		9	47		ANS	90		5
5		:	48		IM	91		:
6		8	49		:	92		3
7		:	50		ST #	93		=
8		7	51		2	94		3
9		:	52		:	95		log
10		6	53		2	96		X
11		:	54		=	97		4
12		5	55		5	98		:
13		=	56		X	99		ANS
14		8	57		5	100		IM
15		÷	58		−	101		:
16		7	59		K	102		
17		:	60		1	103		
18		4	61		:	104		
19		=	62		2	105		
20		9	63		=	106		
21		÷	64		2	107		
22		6	65		√	108		
23		√	66		+	109		
24		:	67		5	110		
25		IF	68		:	111		
26		I	69		2	112		
27		=	70		=	113		
28		K	71		2	114		
29		2	72		log	115		
30		:	73		X	116		
31		1	74		K	117		
32		:	75		2	118		
33		2	76		X	119		
34		:	77		4	120		
35		3	78		:	121		
36		:	79		ANS	122		
37		ST #	80		IM	123		
38		1	81		:	124		
39		:	82		ST #	125		
40		1	83		3	126		
41		=	84		:	127		
42		5	85		3	128		
43		log	86		=	129		

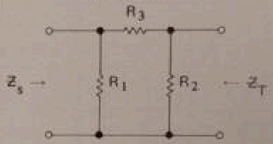
3.44 T AND π ATTENUATORS

T and π attenuators are used to match a high impedance source, Z_s, to a low impedance termination, Z_T, while attenuating the signal between Z_s and Z_T. The T and π attenuator can also provide some desired attenuation between matched source and termination impedances while maintaining matched conditions. This program computes the loss, L_{MIN} in dB that exists for a given Z_s and Z_T when no attenuator network is used. Resistor values R_1, R_2 and R_3 are then computed for an attenuator having a desired loss, L_{DES} in dB greater than L_{MIN} while matching source and termination impedances Z_s must be greater than Z_T.

T Attenuator



π Attenuator



where: Z_s = source impedance
Z_T = termination impedance
$Z_s > Z_T$

The minimum loss, L_{MIN}, in dB of the system without attentuation is given by:

$$L_{MIN} = 10 \operatorname{Log}\left[\left(\frac{Z_s}{Z_T}\right)^{1/2} + \left(\frac{Z_s}{Z_T} - 1\right)^{1/2}\right]^2 \text{ dB}$$

If L_{DES} in decibels is the desired loss of the network, then Attenuation $(ATTN) = 10^{\left(\frac{L_{DES}}{10}\right)}$; $999 > L_{DES} > L_{MIN}$ and:

1. T Attenuator

$$R_1 = Z_s\left[\frac{ATTN+1}{ATTN-1}\right] - R_3 \qquad R_3 = \left[\frac{2}{ATTN-1}\right]\left[ATTN\,(Z_s Z_T)\right]^{1/2}$$

$$R_2 = Z_T\left[\frac{ATTN+1}{ATTN-1}\right] - R_3$$

2. π Attenuator

$$\frac{1}{R_1} = \frac{1}{Z_s}\left[\frac{ATTN+1}{ATTN-1}\right] - \frac{1}{R_3} \qquad \frac{1}{R_3} = \left[\frac{2}{ATTN-1}\right]\left[ATTN\left(\frac{1}{Z_s Z_T}\right)\right]^{1/2}$$

$$\frac{1}{R_2} = \frac{1}{Z_T}\left[\frac{ATTN+1}{ATTN-1}\right] - \frac{1}{R_3}$$

where $10^{-48} < R_1 R_2 R_3 < 10^{48}$

Two programs are available. The first accepts the values of Z_s and Z_l and computes the value of L_{MIN} in dB. A value of L_{DES} is then entered and the program computes the component values for the T attenuator.

The second program computes the component values of the π attenuator given the values of Z_s, Z_l, and L_{DES} in dB.

Program 1 T attenuator components

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of Z_s	ENT
2	ENT	1	Enter the value of Z_l	ENT
3	ANS	2	Displays the value of L_{MIN} (dB)	ANS
4	ENT	3	Enter the value of L_{DES} in dB	ENT
5	ANS	7	Displays the value of R_1	ANS
6	ANS	8	Displays the value of R_2	ANS
7	ANS	9	Displays the value of R_3	
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		3	87		4
2		0	45		÷	88		:
3		:	46		K	89		7
4		1	47		1	90		=
5		:	48		0	91		0
6		2	49		:	92		X
7		=	50		3	93		5
8		0	51		=	94		–
9		÷	52		3	95		9
10		1	53		10^x	96		:
11		:	54		:	97		8
12		3	55		4	98		=
13		=	56		=	99		1
14		2	57		3	100		X
15		–	58		–	101		5
16		K	59		K	102		–
17		1	60		1	103		9
18		:	61		:	104		:
19		2	62		5	105		ANS
20		=	63		=	106		7
21		2	64		3	107		:
22		√	65		+	108		8
23		+	66		K	109		:
24		3	67		1	110		9
25		√	68		÷	111		:
26		:	69		4	112		
27		2	70		:	113		
28		=	71		9	114		
29		2	72		=	115		
30		log	73		3	116		
31		X	74		X	117		
32		K	75		0	118		
33		2	76		X	119		
34		0	77		1	120		
35		:	78		:	121		
36		ANS	79		9	122		
37		2	80		=	123		
38		:	81		9	124		
39		ENT	82		√	125		
40		3	83		X	126		
41		:	84		K	127		
42		3	85		2	128		
43		=	86		÷	129		

Program 2. π Attenuator components

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of Z_s	ENT
2	ENT	1	Enter the value of Z_i	ENT
3	ENT	3	Enter the value of L_{DES} in dB	ENT
4	ANS	7	Displays the value of R_1	ANS
5	ANS	8	Displays the value of R_2	ANS
6	ANS	9	Displays the value of R_3	
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		3	87		K
2		0	45		+	88		1
3		:	46		K	89		÷
4		1	47		1	90		7
5		:	48		÷	91		:
6		3	49		4	92		8
7		:	50		:	93		=
8		0	51		9	94		K
9		=	52		=	95		1
10		K	53		3	96		÷
11		1	54		X	97		8
12		÷	55		0	98		:
13		0	56		X	99		9
14		:	57		1	100		=
15		1	58		:	101		K
16		=	59		9	102		1
17		K	60		=	103		÷
18		1	61		9	104		9
19		÷	62		√	105		:
20		1	63		X	106		ANS
21		:	64		K	107		7
22		3	65		2	108		:
23		=	66		÷	109		8
24		3	67		4	110		:
25		÷	68		:	111		9
26		K	69		7	112		:
27		1	70		=	113		
28		0	71		0	114		
29		:	72		X	115		
30		3	73		5	116		
31		=	74		−	117		
32		3	75		9	118		
33		10^x	76		:	119		
34		:	77		8	120		
35		4	78		=	121		
36		=	79		1	122		
37		3	80		X	123		
38		−	81		5	124		
39		K	82		−	125		
40		1	83		9	126		
41		:	84		:	127		
42		5	85		7	128		
43		=	86		=	129		

General Programs

3.45 ARITHMETIC, GEOMETRIC AND HARMONIC PROGRESSIONS

The program can compute the following terms and sums.

1. For an **arithmetic progression** whose nth term is given by

$$A_n = A_1 + (n-1)\,d$$

it computes A_n given A_1, n and d, and also computes the sum of the first n terms of the progression using

$$S_n = (A_1 + A_n) \times n/2$$

2. For a **geometric progression** whose nth term is given by

$$G_n = G_1\, r^{n-1}$$

it computes G_n given G_1, n and r, and also computes the sum of the first n terms of the progression using

$$S_n = G_1 \left(\frac{1-r^n}{1-r} \right) \qquad \text{It is required that } r \neq 1.$$

3. For an **harmonic progression** it computes the nth term of the progression from

$$H_n = \frac{a}{b + (n-1)\,c}$$

given a, b, c and n and that $a + (n-1)\,c \neq 0$.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	E	Enter 0 for A.P., 1 for G.P. or 2 for H.P.	ENT
2	ENT	0	If A.P. enter A_1 if G.P. enter G_1 or if H.P. enter a	ENT
3	ENT	1	If A.P. enter d, if G.P. enter r, or if H.P. enter b.	ENT
4	ENT	2	If A.P. or G.P. enter n, or if H.P. enter c	ENT
5	*ENT	4	* Only if I = 2, n is entered for a H.P.	ENT
6	ANS	3	nth term of the progression	ANS
7	ANS	4	For A.P. or G.P., S_n is displayed. For H.P., n is displayed.	
8				
9				

PROGRAM LISTING

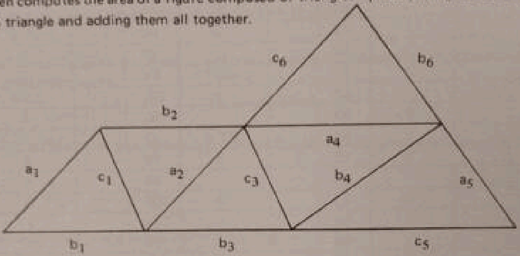
STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44		0	87		5
2		I	45		X	88		:
3		:	46		2	89		GOTO
4		0	47		÷	90		4
5		:	48		K	91		:
6		1	49		2	92		ST #
7		:	50		:	93		3
8		2	51		GOTO	94		:
9		:	52		4	95		ENT
10		9	53		:	96		4
11		=	54		ST #	97		:
12		2	55		2	98		3
13		—	56		:	99		=
14		K	57		3	100		4
15		1	58		=	101		—
16		:	59		1	102		K
17		IF	60		x^y	103		1
18		I	61		9	104		X
19		=	62		X	105		2
20		K	63		0	106		+
21		1	64		:	107		1
22		:	65		4	108		:
23		1	66		=	109		3
24		:	67		1	110		=
25		2	68		x^y	111		0
26		:	69		2	112		÷
27		3	70		—	113		3
28		:	71		K	114		:
29		ST #	72		1	115		ST #
30		1	73		X	116		4
31		:	74		0	117		:
32		3	75		:	118		ANS
33		=	76		5	119		3
34		9	77		=	120		:
35		X	78		1	121		4
36		1	79		—	122		:
37		+	80		K	123		
38		0	81		1	124		
39		:	82		:	125		
40		4	83		4	126		
41		=	84		=	127		
42		3	85		4	128		
43		+	86		÷	129		

3.46 THE AREA OF A TRIANGLE AND OF A FIGURE COMPOSED OF TRIANGLES

This program computes the area, A, of a triangle given the length of its sides using the formulae

$$s = \tfrac{1}{2}(a + b + c)$$
$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

where a, b, c are the lengths of the sides of the triangle.
It then computes the area of a figure composed of triangles by computing the area of each triangle and adding them all together.



USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of a	ENT
2	ENT	2	Enter the value of b	ENT
3	ENT	3	Enter the value of c	ENT
4	ANS	4	Displays the value of A	ANS
5			⋮	
6	ENT	1	Enter the value of a for the final triangle	ENT
7	ENT	2	Enter the value of b for the final triangle	ENT
8	ENT	3	Enter the value of c for the final triangle	ENT
9	ANS	4	Displays the value of A for the final triangle	MJ
10	ANS	5	Displays the value of the sum of the areas of the triangles	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		X	87		
2		ST #	45		0	88		
3		1	46		:	89		
4		:	47		4	90		
5		ENT	48		=	91		
6		1	49		3	92		
7		:	50		√	93		
8		2	51		:	94		
9		:	52		5	95		
10		3	53		=	96		
11		:	54		4	97		
12		0	55		+	98		
13		=	56		5	99		
14		1	57		:	100		
15		+	58		ANS	101		
16		2	59		3	102		
17		+	60		:	103		
18		3	61		GOTO	104		
19		÷	62		1	105		
20		K	63		:	106		
21		2	64		MJ	107		
22		:	65		ANS	108		
23		3	66		5	109		
24		=	67		:	110		
25		0	68			111		
26		−	69			112		
27		3	70			113		
28		:	71			114		
29		3	72			115		
30		=	73			116		
31		0	74			117		
32		−	75			118		
33		2	76			119		
34		X	77			120		
35		3	78			121		
36		:	79			122		
37		3	80			123		
38		=	81			124		
39		0	82			125		
40		−	83			126		
41		1	84			127		
42		X	85			128		
43		3	86			129		

3.47 AREA AND SIDE OF A REGULAR POLYGON

Given the radius of a circle, r, and the number of sides of a polygon, n, this program computes the length, a, of a side of an n-sided regular polygon inscribed in a circle of radius r and the area, S, of the polygon. The formulae used are

$$a = 2r \sin\left(\frac{\alpha}{2}\right)$$

$$S = \frac{nr^2}{2} \sin\alpha$$

$$\alpha = \frac{360^\circ}{n}$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of r	ENT
2	ENT	2	Enter the value of n	ENT
3	ANS	4	Displays the value of a	ANS
4	ANS	5	Displays the value of S	ANS
5			Permits the repetition	
6			of the computation.	
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ST #	44		SIN	87		
2		1	45		÷	88		
3		:	46		K	89		
4		ENT	47		2	90		
5		1	48		:	91		
6		:	49		ANS	92		
7		2	50		4	93		
8		:	51		:	94		
9		3	52		5	95		
10		=	53		:	96		
11		K	54		GOTO	97		
12		3	55		1	98		
13		6	56		:	99		
14		0	57			100		
15		÷	58			101		
16		2	59			102		
17		:	60			103		
18		4	61			104		
19		=	62			105		
20		3	63			106		
21		÷	64			107		
22		K	65			108		
23		2	66			109		
24		:	67			110		
25		4	68			111		
26		=	69			112		
27		K	70			113		
28		2	71			114		
29		X	72			115		
30		1	73			116		
31		X	74			117		
32		4	75			118		
33		SIN	76			119		
34		:	77			120		
35		5	78			121		
36		=	79			122		
37		1	80			123		
38		X	81			124		
39		1	82			125		
40		X	83			126		
41		2	84			127		
42		X	85			128		
43		3	86			129		

3.48 THE SPEED AND HALF-LIFE OF A REACTION

Given the value of the initial concentration, a, and a series of values (x_i, t_i), where x_i is the reduction of concentration (mol/l) after time t_i
this program computes the reaction speed constant, $\underline{k}$, and the half-life of the reaction using

$$k_i = \frac{1}{t_i} \ln\left(\frac{a}{a - x_i}\right)$$

$$\overline{k} = \frac{1}{n} \sum_{i=1}^{n} k_i$$

$$\text{half-life} = \frac{\ln 2}{\underline{k}}$$

and n is the number of observations.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of a	ENT
2	ENT	2	Enter the value of x_1	ENT
3	ENT	3	Enter the value of t_1	ENT
4	ANS	6	Displays the value of k_1	ANS
5			. . .	
6	ENT	2	Enter the value of x_n	ENT
7	ENT	3	Enter the value of t_n	ENT
8	ANS	6	Displays the value of k_n	MJ
9	ANS	9	Displays the value of $\overline{k}$	ANS
10	ANS	0	Displays the value of the half-life	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		:	87		
2		ENT	45		ANS	88		
3		1	46		6	89		
4		:	47		:	90		
5		ST #	48		GOTO	91		
6		1	49		1	92		
7		:	50		:	93		
8		ENT	51		MJ	94		
9		2	52		9	95		
10		:	53		=	96		
11		3	54		8	97		
12		:	55		÷	98		
13		4	56		7	99		
14		=	57		:	100		
15		1	58		0	101		
16		−	59		=	102		
17		2	60		K	103		
18		:	61		2	104		
19		5	62		ln	105		
20		=	63		÷	106		
21		1	64		9	107		
22		÷	65		:	108		
23		4	66		ANS	109		
24		:	67		9	110		
25		6	68		:	111		
26		=	69		0	112		
27		5	70		:	113		
28		ln	71			114		
29		÷	72			115		
30		3	73			116		
31		:	74			117		
32		7	75			118		
33		=	76			119		
34		7	77			120		
35		+	78			121		
36		K	79			122		
37		1	80			123		
38		:	81			124		
39		8	82			125		
40		=	83			126		
41		6	84			127		
42		+	85			128		
43		8	86			129		

3.49 STADIA SURVEYING

This type of computation is widely used in civil engineering, construction and surveying.

Given the stadium multiplication constant, K, the stadium addition constant, C, the height of the surveying instrument, j, the vertical angle , α, (in degrees), the lower stadium wire reading, l_1 , and the upper stadium wire reading, l_2 , this program will compute the horizontal distance between the stadium and the instrument location, S, and the height of the station above the instrument location, H. The formulae used are:

$$H = \frac{1}{2} Kl \sin 2\alpha + C \sin\alpha + j - z$$

$$S = Kl \cos^2\alpha + C \cos\alpha$$

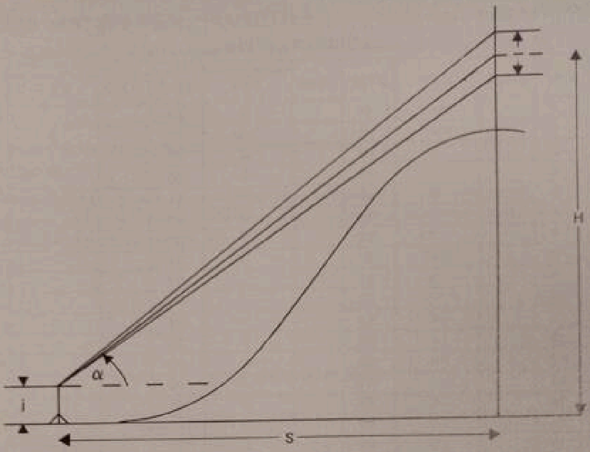
$$l = l_2 - l_1$$

$$z = \frac{l_2 + l_1}{2}$$

To repeat the computation when the stadium constants are the same press MJ, otherwise press START.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter stadium multiplication constant	ENT
2	ENT	2	Enter stadium addition constant	ENT
3	ENT	3	Enter the value of α (degrees)	ENT
4	ENT	4	Enter the value of l_1	ENT
5	ENT	5	Enter the value of l_2	ENT
6	ENT	6	Enter the value of j	ENT
7	ANS	5	Displays the value of S	ANS
8	ANS	6	Displays the value of H	
9				
10				



α
i
s
H

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		3	87		:
2		ENT	45		cos	88		6
3		1	46		X	89		:
4		:	47		1	90		
5		2	48		X	91		
6		:	49		0	92		
7		MJ	50		+	93		
8		ENT	51		5	94		
9		3	52		:	95		
10		:	53		7	96		
11		4	54		=	97		
12		:	55		3	98		
13		5	56		sin	99		
14		:	57		X	100		
15		6	58		2	101		
16		:	59		:	102		
17		0	60		8	103		
18		=	61		=	104		
19		5	62		3	105		
20		—	63		X	106		
21		4	64		K	107		
22		:	65		2	108		
23		4	66		:	109		
24		=	67		6	110		
25		5	68		=	111		
26		+	69		8	112		
27		4	70		sin	113		
28		÷	71		X	114		
29		K	72		0	115		
30		2	73		X	116		
31		:	74		1	117		
32		5	75		÷	118		
33		=	76		K	119		
34		3	77		2	120		
35		cos	78		+	121		
36		X	79		7	122		
37		2	80		+	123		
38		:	81		6	124		
39		5	82		—	125		
40		=	83		4	126		
41		3	84		:	127		
42		cos	85		ANS	128		
43		X	86		5	129		

3.50 MOTION OF A PROJECTILE

The parabolic motion of a projectile starting from the origin with initial velocity V_o and elevation α is described by the equations

$$x = V_o\, t \cos\alpha$$
$$y = V_o\, t \sin d\alpha - ½ gt^2$$

where (x,y) are the coordinates of the projectile at time t and g is the acceleration due to gravity.

This program computes x and y for t from 0 in intervals of 0.1 until y becomes negative, that is, until the projectile lands. The program then terminates displaying the time of flight of the projectile, given by

$$t = \frac{2}{g} V_o \sin\alpha$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	1	Enter the value of V_o	ENT
2	ENT	2	Enter the value of α	ENT
3	ANS	3	Displays the value of x when t = 0.1	ANS
4	ANS	6	Displays the value of y when t = 0.1	ANS
5	ANS	3	Displays the value of x and y for t = 0.1 i	ANS
6	ANS	6	until y becomes negative	ANS
7	ANS	0	Displays the time of flight of the projectile.	
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		X	87		X
2		ENT	45		0	88		K
3		1	46		÷	89		2
4		:	47		K	90		:
5		2	48		2	91		ANS
6		:	49		:	92		0
7		ST #	50		6	93		:
8		1	51		=	94		
9		:	52		4	95		
10		0	53		−	96		
11		=	54		5	97		
12		0	55		:	98		
13		+	56		ANS	99		
14		K	57		3	100		
15		.	58		:	101		
16		1	59		6	102		
17		:	60		:	103		
18		3	61		IF	104		
19		=	62		6	105		
20		1	63		=	106		
21		X	64		K	107		
22		2	65		0	108		
23		cos	66		:	109		
24		X	67		2	110		
25		0	68		:	111		
26		:	69		2	112		
27		4	70		:	113		
28		=	71		1	114		
29		1	72		:	115		
30		X	73		ST #	116		
31		2	74		2	117		
32		sin	75		:	118		
33		X	76		0	119		
34		0	77		=	120		
35		:	78		1	121		
36		5	79		X	122		
37		=	80		2	123		
38		K	81		sin	124		
39		9	82		÷	125		
40		.	83		K	126		
41		8	84		9	127		
42		X	85		.	128		
43		0	86		8	129		

3.51 PERIOD OF OSCILLATION

The period of oscillation of a pendulum is given by

$$T = 2\pi\sqrt{\frac{l}{g}}$$

where T is the period of oscillation in secs
l is the length of the pendulum in cm.
g is the acceleration due to gravity = 980 cms^{-2}

This program computes the value of T given the value of l. The value of π is also entered as data in order to obtain the maximum accuracy possible.

USER INSTRUCTIONS

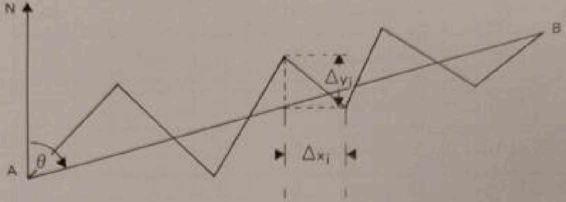
STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	0	Enter the value of l	ENT
2	ENT	1	Enter π	ENT
3	ANS	2	Displays the value of T	
4				
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		ENT	44			87		
2		0	45			88		
3		:	46			89		
4		1	47			90		
5		:	48			91		
6		2	49			92		
7		=	50			93		
8		0	51			94		
9		÷	52			95		
10		K	53			96		
11		9	54			97		
12		8	55			98		
13		0	56			99		
14		:	57			100		
15		2	58			101		
16		=	59			102		
17		2	60			103		
18		√	61			104		
19		:	62			105		
20		2	63			106		
21		=	64			107		
22		2	65			108		
23		X	66			109		
24		1	67			110		
25		X	68			111		
26		K	69			112		
27		2	70			113		
28		:	71			114		
29		ANS	72			115		
30		2	73			116		
31		:	74			117		
32			75			118		
33			76			119		
34			77			120		
35			78			121		
36			79			122		
37			80			123		
38			81			124		
39			82			125		
40			83			126		
41			84			127		
42			85			128		
43			86			129		

3.52 COURSE CALCULATION

After a series of tacking movements, as shown in the diagram, specified by Δx_i, Δy_i for $i = 1 \ldots n$, this program computes the distance of the final position, B, from the initial position, A, as well as its bearing.



The formulae used are

$$\theta = \tan^{-1} \frac{\sum_{i=1}^{n} \Delta y_i}{\sum_{i=1}^{n} \Delta x_i}$$

$$AB = \frac{\sum_{i=1}^{n} \Delta y_i}{\sin \theta} \qquad \text{if } \left| \sum_{i=1}^{n} \Delta x_i \right| < \left| \sum_{i=1}^{n} \Delta y_i \right|$$

$$AB = \frac{\sum_{i=1}^{n} \Delta x_i}{\cos \theta} \qquad \text{if } \left| \sum_{i=1}^{n} \Delta x_i \right| > \left| \sum_{i=1}^{n} \Delta y_i \right|$$

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	9	Enter the value of Δx_1	ENT
2	ENT	0	Enter value of Δy_1	ENT
3	ENT	9	Enter value of Δx_2	ENT
4	ENT	0	Enter value of Δy_2	ENT
5			⋮	
6	ENT	9	Enter value of Δx_n	ENT
7	ENT	0	Enter value of Δy_n	ENT
8	ENT	9		MJ
9	ANS	9	Displays the value of θ	ANS
10	ANS	0	Displays the value of AB	

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		1	87		0
2		ST #	45		:	88		=
3		1	46		IM	89		3
4		:	47		=	90		÷
5		ENT	48		3	91		9
6		9	49		:	92		cos
7		:	50		GOTO	93		:
8		0	51		0	94		ANS
9		:	52		:	95		0
10		3	53		IM	96		:
11		=	54		=	97		SUB #
12		3	55		4	98		0
13		+	56		:	99		:
14		9	57		GOTO	100		IF
15		:	58		0	101		IM
16		4	59		:	102		=
17		=	60		IF	103		K
18		4	61		1	104		0
19		+	62		=	105		:
20		0	63		2	106		2
21		:	64		:	107		:
22		GOTO	65		4	108		3
23		1	66		:	109		:
24		:	67		5	110		3
25		MJ	68		:	111		:
26		5	69		5	112		ST #
27		=	70		:	113		2
28		4	71		ST #	114		:
29		÷	72		4	115		IM
30		3	73		:	116		=
31		:	74		0	117		IM
32		9	75		=	118		±
33		=	76		3	119		:
34		5	77		÷	120		ST #
35		arc	78		9	121		3
36		tan	79		sin	122		:
37		:	80		:	123		I
38		ANS	81		ANS	124		=
39		9	82		0	125		K
40		:	83		:	126		2
41		I	84		ST #	127		:
42		=	85		5	128		
43		K	86		:	129		

3.53 HOURS, MINUTES AND SECONDS

This program accepts a decimal number representing a period of time in hours, and converts it to hours minutes and seconds.

In the same way it will also convert a number of degrees expressed as a decimal number to degrees, minutes and seconds.

USER INSTRUCTIONS

STEP	LAMP	MEM.	INPUT OR OUTPUT	KEY
				START
1	ENT	3	Enter the number of hours as a fraction	ENT
2	ANS	1	Displays the number of hours	ANS
3	ANS	2	Displays the number of minutes	ANS
4	ANS	3	Displays the number of seconds if it is non-zero.	
5				
6				
7				
8				
9				
10				

PROGRAM LISTING

STEP	CODE	KEY	STEP	CODE	KEY	STEP	CODE	KEY
1		MAC	44		2	87		ST #
2		0	45		:	88		5
3		=	46		ST #	89		:
4		K	47		2	90		2
5		3	48		:	91		=
6		6	49		3	92		2
7		0	50		=	93		+
8		0	51		3	94		K
9		:	52		–	95		1
10		ENT	53		K	96		:
11		3	54		6	97		ANS
12		:	55		0	98		1
13		3	56		:	99		:
14		=	57		2	100		2
15		3	58		=	101		:
16		X	59		2	102		
17		0	60		+	103		
18		:	61		K	104		
19		ST #	62		1	105		
20		1	63		:	106		
21		:	64		IF	107		
22		3	65		K	108		
23		=	66		6	109		
24		3	67		0	110		
25		–	68		=	111		
26		0	69		3	112		
27		:	70		:	113		
28		1	71		2	114		
29		=	72		:	115		
30		1	73		5	116		
31		+	74		:	117		
32		K	75		4	118		
33		1	76		:	119		
34		:	77		ST #	120		
35		IF	78		4	121		
36		0	79		:	122		
37		=	80		ANS	123		
38		3	81		1	124		
39		:	82		:	125		
40		1	83		2	126		
41		:	84		:	127		
42		3	85		3	128		
43		:	86		:	129		

SPECIFICATIONS

NORMAL OPERATION

Capabilities

4 basic functions, chain & mixed operations, constant calculations for five functions, powers and reciprocals, automatic accumulation in four functions, direct access to the memory, true credit balance and various kinds of practical calculations.

SCIENTIFIC FUNCTION

Trigonometric/Inverse trigonometric functions, common/natural logarithmic functions, Exponentiations, square roots, reciprocals, sexagesimal/decimal conversion, Pi entry and scientific notation.

Capacity :

	Input range	Output accuracy
Entry/basic operations	10 digit mantissa or 8 digit mantissa plus 2 digit exponent (powers of ten from 10^{99} to to 10^{-99}).	
$\sin x$ /$\cos x$ /$\tan x$	$\|x\| \leqq 1440°$ (8πrad, 1600gra)	±1 in the 8th digit
$\sin^{-1} x$ /$\cos^{-1} x$	$\|x\| \leqq 1$	±1 in the 8th digit
$\tan^{-1} x$	$\|x\| < 1 \times 10^{100}$	±1 in the 8th digit
$\log x$ /$\ln x$	$0 < x < 1 \times 10^{100}$	±1 in the 8th digit
10^x	$\|x\| < 100$	±1 in the 8th digit
e^x	$\|x\| \leqq 230$	±1 in the 8th digit
x^y	$0 < x < 1 \times 10^{100}$	±1 in the 7th digit
$\sqrt{x}$	$0 \leqq x < 1 \times 10^{100}$	±1 in the 10th digit
$1/x$	$\|x\| < 1 \times 10^{100}$, $x \neq 0$	±1 in the 10th digit
°'"	Up to second	±1 in the 10th digit
π	10 digit	

DECIMAL POINT Full floating mode with underflow

NEGATIVE NUMBER

Indicated by the floating minus (–) sign for mantissa.
The minus sign appears in the 3rd column for a negative exponent.

OVERFLOW Indicated by an "E." sign, locking the calculator.

MEMORY 1 independent memory and 10 data memories.

PROGRAM

Number of steps: 127 steps, stored system

Memory: 10 memories for calculation and data totaling plus 1 indirect address memory

Conditional and unconditional jump: max. of 10 jumps possible

Subroutines: max of 10 subroutines, 1 deep

Other functions: Manual jump, multiple assembly of one constant, program writing and check command display, and back-step.

READ-OUT

Zero suppression, Digitron tube panel, and LED for signs

POWER SOURCE

AC: 100, 117, 220 or 240V (± 10V), 50/60 Hz with applicable AC Adaptor

DC: Four AA size Manganese dry batteries (SUM–3) operate about 8 hours continuously.
Four AA size Alkaline dry batteries (AM–3) operate about 19 hours continuously.

USABLE TEMPERATURE

0°C ~ 40°C (32°F ~ 104°F)

DIMENSIONS
34.3mmH x 104mmW x 172mmD (1-3/8"H x 4"W x 7"D)
WEIGHT
364 g (12.8 oz) including batteries

CARE OF YOUR NEW ELECTRONIC CALCULATOR

This calculator is a durable, precision-made instrument which will provide you with years of trouble-free service.

To help ensure this we recommend that the inside of the calculator not be touched. It is also inadvisable to subject the calculator to hard knocks, and unduly strong key pressing.

Extreme cold (below 32° F or 0° C), heat (above 104° F or 40° C) and humidity may also effect the function of the calculator. When you do not use the calculator for a long period, take out the batteries to prevent possible damage from battery leakage. Special care should be taken not to leave dead batteries inside the calculator. Please make sure you switch off the power when you finish your calculations or intend to open the cover to change batteries. Should the calculator need servicing, take the unit to the store where purchased or to a nearby dealer.

CASIO

Casio Electronics Co Ltd, 28 Scrutton Street, EC2A 4TY. Tel: 01-377 9087