Sinclair Cambridge Programmable

Operating Instructions

Sinclair Radionics Ltd
London Rd
St Ives
Huntingdon
Cambs PE17 4HJ
part no. 48584 340

sinclair

See what we mean?

Just two examples of what programmability is and how it can help everyone from scientist to householder.

Now turn to the instruction book for a full explanation of programmability. It tells you how to enter a standard program from the program library (page 18), how to check the program (page 19), how to correct the program (page19), and finally how to write your own programs (page 21).

Sinclair Radionics Ltd,
London Road, St Ives, Huntingdon,
Cambs., PE17 4HJ.
Tel: St Ives (0480) 64646.

sinclair

Part No. 48584-500

READ THIS FIRST!

Your new Cambridge Programmable is really three calculators in one. It's a powerful, straightforward, arithmetic calculator. It's an advanced scientific calculator. And it's a 36-step, keyboard entry programmable calculator.

The instruction book covers all aspects of the calculator in detail. But before reading it, just a couple of examples of programmability. Simply switch on the calculator, turn the page, and away you go.

Convert degrees Centigrade to degrees Fahrenheit

The formula is °C = °F x 1.8 + 32.
This can, of course, be calculated on most non-programmable calculators, but if you had a whole series of conversions to do it would mean at least 10 keystrokes each time. The Cambridge Programmable changes all that.

Try it and see. Enter the programme by pressing these keys.

▲/▼ ▲/▼ 2 0 0 ▲/▼ RUN (Tells the calculator you are about to enter a program.)

× 3 1 ·/EE/- 8 + 3 3 2 =
0 ·/EE/- 2 0 0 (The program.)

C/CE ▲/▼ ▲/▼ 2 0 0 (Tells the calculator you've finished entering the program.)

You've now programmed the calculator to convert any degrees Centigrade to degrees Fahrenheit.

For example
10°C = ?°F
Simply press 1 0 RUN and the calculator shows 50 (50°F)

0°C = ?°F
Press 0 RUN and the calculator shows 32 (32°F)

100°C = ?°F
Press 1 0 0 RUN and the calculator shows 212 (212°F)

And so on. Each time you only need to enter the number of degrees Centigrade and press the RUN button and the calculator does the rest.

Now try another example.

Mortgage repayment

You take out a mortgage for £8,500. The rate of interest is 11%, and you can pay it off over 25 years.

The question is, how much will you have to pay each month?
On a normal calculator this is a very complicated procedure – indeed on many non-programmable calculators it's virtually impossible.

with the Cambridge Programmable all you do is enter the owing program:

▲/▼ ▲/▼ 2 0 0 ▲/▼ RUN (Tells the calculator you are about to enter a program.)

÷ 3 1 0 0 + 2 3 1 =
4 × 0 = ·/EE/- 4 ÷ − 3 1
− ÷ 5 ÷ 0 ÷ ÷ 3 1 2
= 0 ·/EE/- 2 0 0 (The program.)

C/CE ▲/▼ ▲/▼ 2 0 0 (Tells the calculator you've finished entering the program.)

Now enter the interest rate by keying in 1 1 RUN

enter the number of years, 2 5 RUN and finally

the amount 8 5 0 0 RUN

The display immediately shows 84.107. therefore £84.11 per month is what you will have to pay.

And if you want to do a similar calculation but vary the interest rate, the period of the loan, or the rate of interest – no problem.
And certainly no need to re-program.

Let's say the interest rate was 12%, the period of the loan was 20 years, and the amount £10,000. Simply press 1 2 RUN 2
RUN 1 0 0 0 0 RUN and you get the answer 111.579 (£111.58 per month).

Now turn over.

INTRODUCTION

Electronic calculators have revolutionised the lives of business-men, students, engineers and scientists. They've made calculations easier, faster, and more available than anyone could have imagined a few years ago.

Today, there are all kinds of electronic machines available ranging from the simple four function pocket calculator to the expensive specialist desk-top machine. However, none of these approaches the combination of pocketability and versatility provided by a pocket programmable calculator—which makes the Cambridge Programmable a remarkable breakthrough.

Programmability enables you to use a long sequence of operations—either your own or one in a program library—several times without having to enter the sequence each time.

The Sinclair Cambridge Programmable is the world's first truly pocketable calculator with a programming facility. The functions of a powerful scientific machine *plus* programmability are contained within this slim compact format—and it costs no more than ordinary scientific calculators.

In conjunction with its program library, it combines all the advantages of a variety of specialist calculators: financial, statistical, mathematical, engineering, etc. You need no knowledge of programming to use these facilities, since easy step-by-step explanations at the back of this book are provided.

At the same time, the simply detailed explanations at the back of this book provide, on the one hand, a valuable teaching aid for those who want to learn programming and, on the other, an invitation to more sophisticated programmers to explore the vast possibilities of this first ever pocket programmable.

PROGRAMMABLE KEYBOARD/VOCABULARY

ON/OFF SWITCH—top right hand corner

DECIMAL POINT AND EXPONENT ENTRY
•/EE/−

TEN DIGITS AND =

ARITHMETIC OPERATORS
+ − × ÷

CONVENIENCE FUNCTIONS (yellow)
Act on result of previous calculation
x^2 $^1/x$ $-x$ $2x$

FUNCTIONS (Basic and program)
Act on display contents

Upper case	**Lower case**
sin	arcsin
cos	arccos
tan	arctan
ln x	e^x
rcl (memory)	MEx (memory)
()	R▶D
$\sqrt{x}$	
sto (memory)	
ChN	D▶R
	+/−

FUNCTIONS WITHIN PROGRAM ONLY

go if neg	▼ (lower case)
stop	# (number)

PROGRAM AND CONTROL INSTRUCTIONS

RUN	learn
step	go to

NOT USED WITHIN PROGRAM

▲/▼ (shift key)	C/CE

CHECK SYMBOLS (blue)
A E F G • −

CONTENTS

This booklet describes the functions of the Cambridge Programmable, explains how to use it, and gives a series of examples of typical calculations. We suggest you work through each example with your calculator, as this will cover all the basic functions of the machine. The text can be referred to where necessary for detailed explanation.

If you have not used a pocket calculator before, we suggest you familiarise yourself with the basic machine before working through the section on programming. You will then be able to use the calculator straight away for simple calculations, and know where to look for reference, when you need to handle complex calculations.

FUNCTIONS AND FEATURES

CAPABILITY

36-step keyboard entry program facility

Basic arithmetic operators + − × ÷

Algebraic logic on all arithmetic operators

Sign change +/−

Memory

Brackets

Six trigonometric functions

sin	arcsin
cos	arccos
tan	arctan

Operation in radians

Radian/degree and degree/radian conversion

Natural logarithms ln$x$ and antilogarithms e^x

$\sqrt{x}$ x^2 $1/x$ $-x$ $2x$

Floating point and scientific notation

Automatic clearing when first switched on

Indicators

Upper and lower case functions
Error condition

DESIGN FEATURES

Three function keyboard—upper and lower case operation

The keyboard layout is very simple: most keys have two or more functions, instantly selectable. A very wide range of functions (including programming) is available from just 19 keys.

The basic keyboard is that of a normal arithmetic calculator,

operating with algebraic logic. The designations for this keyboard are printed in the centre of each key.
The same keys operate an upper and a lower case keyboard. The designations for these are printed above and below the keys respectively. The upper and lower case keyboards are brought into operation by pressing the shift key [▲/▼] the appropriate number of times (see page 13).

FLOATING POINT/SCIENTIFIC NOTATION

The Programmable can present results in floating point (arithmetic) format or scientific (mantissa and exponent) notation. Numbers can be entered using either notation (see page 7).
The largest number that can be displayed in floating point format is 99,999,999 yet the programmable can handle numbers as large as 9.9999999×10^{99}.
Calculation results are accordingly presented between 0.001 and 99,999,999 in floating point format and in scientific notation for results outside that range.
If a result is displayed in scientific notation, the operation sequence

[▲/▼] [3] (ChN)

will cause the eight most significant digits to be displayed and the exponent to be suppressed while being stored internally; a further operation of the same sequence will cause the display to revert to scientific notation.

INDICATORS

F — Upper case function in operation. Results after [▲/▼]

C — Lower case function in operation. Results after [▲/▼] [▲/▼]

E — Error caused by inadmissible operation or overflow.

The indicators are displayed at the left of the display.

POWER SUPPLY

The Cambridge Programmable is a battery operated portable calculator which runs from its internal battery. Disposable Alkaline batteries such as Mallory MN1604 should be used.
For long periods of use, especially in program mode, we recommend that you operate the calculator with the Sinclair Mains Adaptor. Use of any other make of adaptor invalidates the guarantee (see page 27).

NUMBER ENTRY

The programmable accepts numbers in either floating point or scientific notation.

The key [·/EE/-] is used during number entry as follows:

First press	Enters decimal point	
Second press	Prepares for exponent entry	Scientific notation
Third press	Changes sign of exponent	
Subsequent presses	No effect	

Floating point

Key in the digits of the number and the decimal point, if any, up to a maximum of eight digits. The machine will ignore any subsequent digits.

To enter a negative number, press [▲/▼] [▲/▼] [0] (+/−) after entering the digits. The negative sign appears on the left of the display.

eg to enter −32.65 press

[3] [2] [·/EE/-] [6] [5] [▲/▼] [▲/▼] [0] (+/−)

Scientific notation

Scientific notation is a method of expressing numbers in two parts—mantissa and exponent.
The MANTISSA consists of the most significant digits of the number with the decimal point placed after the first digit.
The EXPONENT is the power of 10 by which the mantissa must be multiplied.

Examples

Floating point	Scientific notation Mantissa	Exponent
123,400,000	1.234	08
12345	1.2345	04
0.000123	1.23	−04

Scientific notation enables the programmable to operate on very large or very small numbers, ranging from 10^{-99} to 9.9999999×10^{99}. To enter a number in scientific notation:

1. Enter the digits of the mantissa with the decimal point after the first digit. Press [·/EE/] to enter the decimal point.
2. If the number is negative, press [▲/▼] [▲/▼] [0] (+/−)
3. Press [·/EE/] again to prepare the calculator for exponent entry.
4. Enter the digits of the exponent. (Note: if an error is made simply enter the new digits required—the previous digits will be overwritten.)
5. Finally press [·/EE/] once more if the exponent is negative.

The exponent may take any value from −99 to +99 and appears on the right-hand side of the display.

Clear display after each ex

Examples

To enter	Press	Display shows
123,400,000	[1] [·/EE/] [2] [3] [4]	1.234
	[·/EE/]	1.2340 00
	[8]	1.2340 08
−12345	[1] [·/EE/] [2] [3] [4] [5]	1.2345
	[▲/▼] [▲/▼] [0]	−1.2345
	[·/EE/−] +/−	−1.2345 00
	[4]	−1.2345 04
0.000123	[1] [·/EE/−] [2] [3]	1.23
	[·/EE/]	1.2300 00
	[4]	1.2300 04
	[·/EE/]	1.2300−04

Note: Although the decimal point normally appears after the first digit of the mantissa, this is not mandatory on number entry.
eg 123,400,000 can be expressed as 1.234×10^8 or 1234×10^5 and may be entered if you wish as

[1] [2] [3] [4] [·/EE/] [·/EE/] [5]

However, the machine will normalise the number to the form 1.2340 08 when the next arithmetic operator is pressed.

Display

The programmable will automatically present results in 8-digit floating point format within the range 0.001 to 99,999,999. Outside that range the result will be presented in scientific notation with a 5-digit mantissa and 2-digit exponent.

Entering the sequence [▲/▼] [3] (ChN)

when a result is displayed in scientific notation will recover the sixth, seventh and eighth significant digits of the mantissa but will display the decimal point after the first digit. The exponent and the location of the decimal point is retained by the machine internally and can be recovered by changing notation again.

CLEARING THE MACHINE

When the machine is first switched on, all functions and registers are clear and the display shows 0.
To clear the display after a calculation terminated with [=], press [C/CE] once. To clear the display after a number in the middle of a calculation, press [C/CE] twice.
After a number has been entered*, one depression of the [C/CE] key clears the number and the previous arithmetic operation, if any. The calculation can therefore be continued by pressing a new operator and number.

eg 3 + 5 =	Press	Display
You enter	[3]	3
3 + 4	[+] [4]	4
in error	[C/CE]	3
	[+] [5]	5
	[=]	8

After pressing [C/CE] in the middle of a calculation to correct a wrongly entered number, remember to re-enter the *previous operator* before entering the correct number.

At all other times*, pressing [C/CE] clears the machine (except program store and memory), ie **after an operator or function.**

* To correct a wrong number entry within brackets, close the brackets, press [C/CE] once and recommence entry with the operator immediately before the brackets. Any attempt to use [C/CE] within brackets may create an error if the calculation is continued.

OVERFLOW

If the result of a calculation is outside the range of the machine, an error indicator E will normally appear in the display and will usually appear on subsequent calculations until [C/CE] is pressed to clear the machine.

SIMPLE ARITHMETIC

The Programmable uses algebraic logic. This means that calculations using the basic operators + − × ÷ and = can be entered exactly as written. Remember to use [▲/▼] [▲/▼] [0] for negative numbers.
+/−

eg 5 + (−3) = is entered as

[5] + [3] [▲/▼] [▲/▼] [0] = Result 2.
+/−

A calculation can be completed by either pressing the next arithmetic operator or using [=]. If [=] has been used, the result displayed can be used as the start of the next calculation, in which case the next entry must be an arithmetic operator. If a number is entered after [=], it automatically supersedes the displayed result.

Example

Press	Display	
[1] [7] [·/EE/−] [2]	17·2	
[+] [3]	3	
[×]	20.2	(17.2 + 3 = 20.2)
[5] [·/EE/−] [4]	5·4	17.2 + 3 × 5.4
[=]	109·08	Result
[÷] [5] [9]	59	÷ 59
[=]	1.848813	Result
[2] [7]	27	New entry
[−] [1] [8] [·/EE/−] [3]	18.3	27–18.3
[=]	8.7	Result
[C/CE]	0	Clear

CONVENIENCE FUNCTIONS

x^2 $^1/x$ $-x$ $2x$

When an arithmetic operator + − × ÷ is followed by a number, it is interpreted in the usual way.
When followed by another operator or by =, the alternative interpretations printed at the bottom left of the key are used.

eg			
x	☒	⊟	Result x^2
x	÷	+	Result $^1/x$
x	+	×	Result $2x$
x	−	=	Result $-x$

Since an arithmetic operator completes the previous calculation, the function will act on the *result* and not only on the previous number entry.

eg **Press**	**Display**	
[3] [×] [4] [÷]	12	
[=]	0.0833333	(1/12)
[2] [+] [5] [×]	7	
[×]	49	(7^2)
[C/CE] [1] [5] [÷] [3] [−]	5	
[+]	−5	
[C/CE] [5] [−] [2] [+]	3	
[=]	6	(2×3)

(If you wish to calculate, say, 3×4^2, rewrite as $4^2 \times 3 =$ and press [4] [×] [×] [3] [=]).
Closing brackets after an operator has a similar effect (see page 16).

Note: The sequence
number operator operator operator . . . .
causes the machine to interpret the convenience function repeatedly.

eg **Press**	**Display**
[2] [×] [×]	4
[×]	16
[×]	256 etc

Note: Although the convenience functions x^2 $^1/x$ $-x$ $2x$ are printed on the keyboard below the keys $+$ $-$ $\times$ $\div$, they do not operate as lower case functions (see next section).

UPPER AND LOWER CASE KEYBOARDS

The shift key [▲/▼] gives access to the upper and lower case functions marked on the keyboard.
Pressing [▲/▼] causes an ⌈ to be shown on the left of the display until the next key is pressed and gives access to the upper case functions.
Pressing [▲/▼] [▲/▼] causes a ⌊ to be shown on the left of the display until the next key is pressed and gives access to the lower case functions.
If [▲/▼] is pressed a third time, its effect is cancelled. This cycle can therefore be used to correct the shift key if pressed in error.

TRIGONOMETRIC, LOG AND SQUARE ROOT FUNCTIONS

sin	arcsin	$\ln x$
cos	arccos	e^x
tan	arctan	$\sqrt{x}$

These act on the number which is already in the display, and so must be entered after the number to which they refer. The calculator operates in radians.

Examples

Press	**Effect**
[·/EE/] [4]	.4
[▲/▼] [7] (sin)	sin
[▲/▼] [▲/▼] [9] (arctan)	arctan

Natural antilogarithms are calculated using the sequence

[▲/▼] [▲/▼] [4] (e^x)

The calculator may take a few seconds to work out some functions. During this time the display will be blank. Multiple expressions using the upper and lower case functions may be calculated, but it is important to remember that the function keys are pressed after the entry or result to which they refer.

eg to calculate $\sqrt{\ln 3.5}$ the sequence is 3.5 [▲/▼] [4] (ln x) [▲/▼] [1] ($\sqrt{x}$)

IMPORTANT: Arithmetic calculations must be completed with [=] before being acted on by upper or lower case functions.

eg ln(0.36 + 5.2) press

[0] [·/EE/] [3] [6] [+] [5] [·/EE/] [2] [=] [▲/▼] [4] (ln x)

Note
All these functions act directly on the display contents and do **not** complete a previous calculation. So, for example,

.4 + .5 [▲/▼] [7] (sin) gives sin .5 not sin .9

To calculate sin (.4 + .5) the [=] key must be used before the sin function.
To calculate .4 + sin .5 the expression must be rewritten as sin .5 + .4.

Degree/Radian Conversion
R→D converts radians to degrees and is operated by the sequence
[▲/▼] [▲/▼] [6] (R▸D)
D→R converts degrees to radians and is operated by the sequence
[▲/▼] [▲/▼] [3] (D▸R)
Both conversions act on the number already in the display.

Ranges of functions

sin, cos, tan	$-1.57 < x < 1.57$
arcsin	$-1 \leqslant x \leqslant 1$
arccos	$0 \leqslant x \leqslant 1$
arctan	any x
ln	$x > 0$
e	$-227.95 \leqslant x \leqslant 230.25$

Outside these ranges, very long execution times are possible, leading to incorrect results. There is some loss of accuracy on tangent towards ±1.57 radians.

BRACKETS
Brackets are used in the same order as you would write them, but remember that upper and lower case functions are entered after the number or expression on which they act.

Example

$$\ln\left(\frac{4+2}{7-2}\right)$$

Press	Display
[4] [+] [2] [÷]	6
[▲/▼] [6] (() [7] [−] [2] [▲/▼] [6] ())	5
[=]	1.2
[▲/▼] [4] (ln x)	0.182316

Do not attempt to enter a new number immediately after closing brackets (this should not in any case be necessary).
Note: Pressing = does not close brackets

Special use of brackets
Opening brackets causes the previous result just calculated to be entered within the brackets until a new number is entered. This is particularly useful within a program, as it saves steps.

Examples

$2 \div (3 + 4) =$	calculates $2 \div 7$
$2 \div (+ 4) =$	calculates $2 \div (2 + 4)$
$5 \times (4\sqrt{x}) =$	calculates $5 \times \sqrt{4}$
$5 \times (\sqrt{x}) =$	calculates $5 \times \sqrt{5} = 5^{3/2}$

Convenience functions

x^2 $^1/x$ $-x$ $2x$

Closing brackets immediately after an arithmetic operator causes the calculator to use the alternative interpretations (as on page 12).

eg **Press**	**Display**	
$\boxed{3}\ \boxed{\times}\ \boxed{\blacktriangle/\blacktriangledown}\ \overset{(}{\boxed{6}}\ \boxed{5}\ \boxed{+}\ \boxed{\blacktriangle/\blacktriangledown}\ \overset{)}{\boxed{6}}$	10	ie 2×5
$\boxed{=}$	30	3×10

MEMORY

$\boxed{\blacktriangle/\blacktriangledown}\ \overset{\text{sto}}{\boxed{2}}$ writes contents of display into memory—previous contents are lost

$\boxed{\blacktriangle/\blacktriangledown}\ \overset{\text{rcl}}{\boxed{5}}$ writes contents of memory into display and retains memory contents

$\boxed{\blacktriangle/\blacktriangledown}\ \boxed{\blacktriangle/\blacktriangledown}\ \underset{\text{MEx}}{\boxed{5}}$ exchanges contents of display and memory

To clear the memory, either switch off machine (this will also clear program and display contents), or clear display and exchange memory and display.
The memory functions may be used after arithmetic operators as part of a calculation. The result of the calculation will be displayed but the memory function will also operate, as shown below.

1. **rcl** (Memory recall)
   The sequence **number operator rcl equals**
   will display the result of the calculation and the memory contents will be retained.
   eg $\boxed{2}\ \boxed{+}\ \boxed{\blacktriangle/\blacktriangledown}\ \overset{\text{rcl}}{\boxed{5}}\ \boxed{=}$ adds 2 and memory contents
2. **ME*x*** (Memory exchange)
   Suppose the memory contains 2 initially.
   The sequence
   $\boxed{7}\ \boxed{\times}\ \boxed{\blacktriangle/\blacktriangledown}\ \boxed{\blacktriangle/\blacktriangledown}\ \underset{\text{MEx}}{\boxed{5}}\ \boxed{=}$ displays the result 14 (7×2) **and** stores the number 7 in the memory

3. **sto** (Memory store)
   The sequence **number operator sto equals**
   will cause the machine to store the number and display the result of the calculation using the stored number. Any previous contents of the memory are lost.

   eg $\boxed{3}\ \boxed{\div}\ \boxed{\blacktriangle/\blacktriangledown}\ \overset{\text{sto}}{\boxed{2}}\ \boxed{=}$ divides 3 by itself leaving 1 in display **and** stores 3 in memory
   $\boxed{6}\ \boxed{-}\ \boxed{\blacktriangle/\blacktriangledown}\ \overset{\text{sto}}{\boxed{2}}\ \boxed{=}$ subtracts 6 from itself leaving 0 in display **and** stores 6 in memory. This is of use in a program as a quick way of clearing the display.

Similarly,

Sequence	**Display**	**Memory**
4 + sto =	8	4
3 × sto =	9	3

PROGRAMMING
PROGRAM LIBRARY

The program library which is available for use with this machine has been designed to cover a wide field of financial, scientific and engineering disciplines. The formulae cover many general applications and in many cases, a standard program can be adapted to solve a particular problem.

A sample booklet of the programs is included with the calculator. Compare the example in the booklet while you read the following instructions.

The programs are arranged as far as possible in a standard format:

1. The **formula** to which the program relates.
2. The **diagram or explanation** of symbols where required.
3. The **program:**
   This is written as a series of keystrokes in the **first** column on the right-hand side of the page; where a key has more than one function, the relevant function is printed as the keystroke. eg the keystroke [8] (cos / arccos) may appear as 8, cos or arccos.
   Notes: The symbol ▼ within a program always refers to the key [▼/EE/]; the symbol # refers to [3] (ChN/#); the abbreviation g.i.n. is "go if neg" and so refers to the key [1] (go if neg).
   The **second** and **third** columns on the right-hand side of the page contain respectively the check symbol and step number for each keystroke (see section on checking the program).
4. The **execution** sequence written as a series of variables and/or keystrokes with [RUN] Results *displayed* are printed in gold.

ENTERING THE PROGRAM

To enter a program into the calculator:

1.	Press [▲/▼] [▲/▼] [2] (go to) [0] [0]	Display shows step programmed at 00 in check symbol form as described below.
2.	Press [▲/▼] [RUN] (learn)	No change in display.
3.	Press the sequence of keys for the program as shown in the first column of the program page.	At each stage the step about to be overwritten is displayed. When the machine is first switched on, every step is zero.
4.	Press [C/CE]	Normal number display is resumed.
5.	Press [▲/▼] [▲/▼] [2] (go to) [0] [0]	The step programmed at 00 will be displayed.

CHECKING THE PROGRAM

Each of the programs in the library is shown in check symbol form in the second column on the right-hand side of the page.

Press [▲/▼] [C/CE] (step) repeatedly, and at each stage the check symbol will appear on the left of the display with the step number on the right.

eg	A.0000	03
	check symbol	step number

After stepping through the program, before execution, press

[▲/▼] [▲/▼] [2] (go to) [0] [0]

Finally, press [C/CE] and the program is ready for use.

CORRECTING THE PROGRAM

If the check symbol for a particular step number is not as indicated in the last two columns of the program page:

1. Press [▲/▼] [▲/▼] [2] (go to) followed by the step number if the appropriate step number is not already displayed.
2. Press [▲/▼] [RUN] (learn)
3. Enter the correct keystroke. The display will then show the next step in the program. If this is also incorrect, enter the correct keystroke. At each stage, the step about to be overwritten will be displayed.

4. When correction has been completed, press [C/CE] Any step which has not been overwritten will not be affected.
5. Press [▲/▼] [▲/▼] [2 go to] [0] [0]

NOTE: To restore normal use of the calculator after entering or checking the programme, press [C/CE].

RUNNING THE PROGRAM

Press the sequence of keys as shown in the program library in the Execution Sequence. Results *displayed* are printed in gold.

INTERRUPTING THE PROGRAM

To interrupt the execution of a program, press the sequence

[÷] [C/CE] [▲/▼] [▲/▼] [2 go to] [0] [0]

This facility is useful when, for example, an error in programming has led to an infinite loop which can be corrected without altering the program completely.

WRITING YOUR OWN PROGRAM

The calculator is able to remember a sequence of up to 36 steps. Such a sequence is referred to as a program. Once a program has been stored in the calculator, it can be executed repeatedly using different data. A stored program does not affect the use of the machine as an ordinary calculator.

The program library available for this calculator contains a wide variety of formulae in general use in finance, science and engineering, and many of these can be adapted for specific problems. After practising with these programs, you may like to write your own, and blank forms are supplied in the library for this purpose.

The three keys [▲/▼], [C/CE] and [RUN] may not be used **within** a program but are used before and after a program as follows:

[learn RUN] The sequence [▲/▼] [learn RUN] will put the machine into "learn" mode. This operation is required before **entering** a new program.

[step C/CE] The sequence [▲/▼] [step C/CE] is used to "step" through a program when **checking** that it has been entered correctly.

[RUN] The keystroke [RUN] is used to **execute** a program.

PROGRAM KEYS

Sixteen of the keys on the machine may be used within a program. The four arithmetic operators + − × ÷ and = are used in the normal way.

The other eleven keys (0 to 9 and ·/EE/−) may, where possible, be interpreted in three ways:

1. UPPER CASE FUNCTIONS
   Within a program, the machine will normally use the **upper case** interpretation of a keystroke.

2. NUMBER ENTRY
   To interpret the **face value** of a keystroke, and therefore enter a number, the keystroke [ChN/# 3] must precede number entry.

   It is read as # and causes subsequent keystrokes to be interpreted as normal case until the next arithmetic operator

(+ − × ÷) or = is pressed, when upper case interpretation of subsequent keystrokes will be restored.

3. LOWER CASE FUNCTIONS

The keystroke [·/EE/−] used within a program is read as ▼ and causes the machine to interpret the next keystroke in its **lower case** function. Subsequent keystrokes will be unaffected.

Examples **Keystrokes**	**Interpretation within a program**
[4]	ln x
[3] [4]	# 4
[·/EE/−] [4]	▼ e^x

(Within a program, [3] is always interpreted as #, not ChN.)

Operation	**Keystrokes required**
sin	[7]
arcsin	[·/EE/−] [7]

Number entry	**Keystroke sequence**
2.7	[3] [2] [·/EE/−] [7]
	[=] (or some other arithmetic operator)

All constants must be followed by + − × ÷ or = to restore the machine to upper case interpretation. Numbers may be entered in either floating point or scientific notation.

Special program steps

[2] go to — Within a program, the sequence [·/EE/−] [2] followed by a 2-digit number between 00 and 35 inclusive, will set the program step counter to the given number (which will not be displayed as it is entered). The keystroke programmed at that step number will be displayed in check symbol form.

eg [·/EE/−] [2] [1] [2]

If the machine encounters this sequence in a program, it will transfer to step 12, and execute steps 12, 13, 14 etc.

[1] go if neg — Within a program, the sequence [·/EE/−] [1] followed by a 2-digit number between 00 and 35 inclusive, tests the sign of the result or number in the machine and, if this is negative, transfers to the step number indicated by the two digits. If the result or number is positive or zero, there is no transfer.

[0] stop — Within a program, this keystroke enables results to be displayed and/or any number or operation, or sequence of numbers or operations, to be entered. There **must** be at least one "stop" in the program, to prevent the machine looping infinitely around the program and displaying no results.

CHECK SYMBOLS

To check that a program has been loaded correctly, press [▲/▼] [C/CE] (step) repeatedly. The machine will step through the program and at each stage the step programmed will be indicated by the check symbol at the left of the display.

The check symbols corresponding to the various keystrokes are shown in the following table:

Keystroke	**Check Symbol**
1, $\sqrt{x}$, go if neg	1
2, sto, go to	2
3, #, D ▶ R	3
4, ln x, e^x	4
5, rcl, MEx	5
6, ( ), R ▶ D	6
7, sin, arcsin	7
8, cos, arccos	8
9, tan, arctan	9
0, stop	0
·/EE/−, ▼	A
=	—
+	E
−	F
×	.
÷	G

For example, F.0000 06
means that program step 06 contains the keystroke [−],

The check symbols for the various keystrokes are also shown on the keyboard; the check symbol for each of the digit keys is the digit on the key, while for the other keys the check symbols are shown on the keyboard at the lower right of the key. The keys [▲/▼] [C/CE] [RUN] cannot be used as program steps and so do not have any check symbols.

To check the keystroke programmed at any particular step number, press [▲/▼] [▲/▼] [2] (go to) followed by the step number, a 2-digit number between 00 and 35 inclusive. The check symbol and step number will be displayed. Subsequent use of [▲/▼] [C/CE] (step) will cause the calculator to step through the program from that point.

It will be necessary to clear the machine and reset to the start of the program before running:

[C/CE], [▲/▼] [▲/▼] [2] (go to), [0] [0]

USEFUL PROGRAM STEP SEQUENCES

To halve x	/÷/+/÷/=/
To cube x	/×/(/×/)/=/
To subtract a from b entered later	/a/−/+/b/=/
To square x	/×/=/
To double x	/+/=/
To find x^4	/×/×/=/
To find $4x$	/+/+/=/
To find $4x^2$	/+/×/=/
To find $x^{3/2}$	/×/(/√/)/=/

STEPS IN WRITING THE PROGRAM

1. Write the sequence of keystrokes for the formula as it would be calculated.
2. Where a number occurs, precede the number with [3] (#) and, if there is no following arithmetic operation, add = after the number.
3. Where a double shift occurs, replace it with [·/EE/] (lower case).
4. Omit all single shifts (upper case).
5. Where a variable or sequence of operations is to be entered, except at the beginning of the program, insert [0] (stop).
6. Ensure that there is at least one [0] (stop) eg where a result is to be displayed, otherwise the program will loop infinitely and never display any results. There should normally be a [0] (stop) at the end of the program.
7. Either make up the total number of steps to 36 by inserting [=] [=] ...., OR insert the sequence [·/EE/] [2] (go to) [0] [0] at the end of the program.

Example

$$e^{(5a^2 + b^2)}$$

1. Sequence of keystrokes

   a [×] [×] [5] [+] [▲/▼] [6] (() b [×] [▲/▼] [6] ()) [=] [▲/▼] [▲/▼] [4] (e^x)

2. Insert [3] (#) before the number.
3. Replace [▲/▼]· [▲/▼] with [·/EE/−].
4. Omit [▲/▼]

   a [×] [×] [3] (#) [5] [+] [6] (() b [×] [6] ()) [=] [·/EE/] [4] (e^x)

5. Insert "stop" in place of variables, except at beginning.
6. Add "stop" for reading of result.

   [×] [×] [3] (#) [5] [+] [6] (() [0] (stop) [×] [6] ()) [=] [·/EE/] [4] (e^x) [0] (stop)

   ↑ to enter b — ↑ to read result, and enter new a

7. Add "go to 00".

/EE/ 2 0 0
(go to)

Rewritten in sequence form, the program for $e^{(5a^2 + b^2)}$ is

/×/×/#/5/+/(/stop/×/)/=/▼/e^x/stop/▼/go to/0/0/

Using one of the blank forms provided in the program library, write the program with the check symbols in column form.

EXECUTION SEQUENCE

1. Any variable before the program eg 'a' above.
2. RUN
3. Any variable or operation sequence required at the first stop 0
4. RUN
5. As 3, etc. at next stop 0, and so on.
6. Read result at last stop 0.
7. Enter new variable as at 1.

Example

The execution sequence for the program for $e^{(5a^2 + b^2)}$ is as follows:

/a/RUN/b/RUN/Result displayed/
/new a/ etc.

The procedures for entering, checking, correcting and running the program are as described on pages 18 and 19.

Service and Guarantee

Your calculator is fully guaranteed from the date of purchase against defects in materials or workmanship for the period specified on the enclosed card.

Should any fault develop during this period, it will be **repaired** or **replaced** (at manufacturer's option) without charge to the owner, if it is returned, carefully packed and postage pre-paid – preferably by registered or recorded delivery – direct to Sinclair Radionics. **Enclose** a letter clearly stating your name and address, the date and place of purchase, and the nature of the fault. The guarantee is invalid if the calculator has been damaged by accident, unreasonable use, neglect, or improper service.

A standard service charge will be made for repairs outside the guarantee period.

Before returning your calculator carefully re-check the instructions and also check that the battery does not need replacing. Please retain the battery.

U.K.	U.S.A.
Sinclair Radionics Ltd.,	Sinclair Radionics Inc.,
Calculator Service Department,	Galleria, 115 East 57th Street,
London Road, St. Ives,	New York 10022.
Huntingdon, Cambs.	
PE17 4HJ.	

Outside the U.K. and U.S.A. contact your local Sinclair dealer or distributor. Please use the space below to record the relevant details for your reference.

Date of purchase ______________________

Place of purchase ______________________

Owner's name and address ______________________

____

London Rd
St Ives
Huntingdon
Cambs PE17 4HJ
part no. 48584 430

sinclair

Sinclair Cambridge Programmable

Works out mortgage repayments
Solves quadratic equations
Calculates linear regression
Helps design a twin-T filter
Plays a lunar landing game!

USERS LIBRARY

We've written programs in the Sinclair Program Library to cover a very wide variety of subjects but we'd very much like to hear about any other interesting programs you've written for your Sinclair Cambridge Programmable. By sending in your own program you will become a member of the Sinclair Programmable Users' Library: we'll keep you in touch with news on the other programs in the Users' Library so you can get even better use of your Sinclair Cambridge Programmable.

Send your programs to Users' Library, Sinclair Radionics Limited, London Road, St Ives, Huntingdon, Cambs PE17 4HJ.

To solve hundreds of problems in finance, mathematics, statistics, physics, engineering and electronics, we've written 294 programs specially for the Sinclair Cambridge Programmable. There are 12 samples in this booklet – the rest are all in the Sinclair Program Library.

Before you try any of the programs, familiarise yourself with the calculator by working, calculator in hand, through the Instruction Booklet enclosed. You'll then be able to use the programs quickly and easily.

Remember these are only sample programs reproduced half size – the full Sinclair Program Library is available from Sinclair Radionics Limited, London Road, St Ives, Huntingdon, Cambridgeshire PE17 4HJ, for £1.95 per volume, or £4.95 for all four volumes.

Whatever your speciality, the program library will make the Sinclair Cambridge Programmable the specialist calculator for you!

Day of the week of Christmas Day (program on facing page)

Entering the program

Press	Display
▲▼ ▲▼ 2 0 0 (go to)	0.0000 00
▲▼ RUN (learn)	0.0000 00

Now press the sequence of keys in the program as shown in the first column on the facing page.

Press	Display
X	0.0000 01
3 (ChN/#)	0.0000 02
1	0.0000 03
•/EE/–	0.0000 04
.	.
.	.
.	.
.	.
=	0.0000 34
0 (stop)	0.0000 35
=	.0000 00

The last step has brought you back to step 00 which shows the check symbol for X (the first step) i.e. . on the left of the display.

As you are already at step 00 there is no need to press

▲▼ ▲▼ 2 0 0 (go to)

but you need to do this if you finish at any other step number.

DAY OF THE WEEK OF CHRISTMAS DAY (1900 – 2099)

X	·	00
#	3	01
1	1	02
·	A	03
2	2	04
4	4	05
9	9	06
6	6	07
–	F	08
#	3	09
2	2	10
6	6	11
3	3	12
1	1	13
+	E	14
#	3	15
7	7	16
+	E	17
▼	A	18
gin	1	19
1	1	20
5	5	21
(	6	22
–	F	23
+	E	24
#	3	25
1	1	26
=	–	27
▼	A	28
gin	1	29
2	2	30
4	4	31
)	6	32
=	–	33
stop	0	34
=	–	35

Execution:

year (in full) / RUN / day as a number

where 1 = Sunday
2 = Monday, etc

Checking the program

Press	Display
	.0000 00
▲▼ C/CE (step)	3.0000 01
▲▼ C/CE (step)	1.0000 02
▲▼ C/CE (step)	A.0000 03
.	.
.	.
.	.
▲▼ C/CE (step)	0.0000 34
▲▼ C/CE (step)	–.0000 35

At each step, the check symbol on the left of the display should correspond with the check symbols shown in the second column on the program.

If you entered the program correctly, press

▲▼ ▲▼ (go to) 2 0 0

then C/CE and you are ready to execute the program.

If you made an error at any stage in the program, read the section on correcting the program on page 19 of the instruction booklet.

Executing the program

Example

Press	Display
1 9 7 7	1977
RUN	1

i.e. Christmas Day in 1977 falls on a Sunday.

BALANCE OUTSTANDING ON A MORTGAGE

Given:

Amount of original mortgage
Monthly repayment
Number of years since mortgage was originally taken out
Rate of interest

Finds:

Balance

Execution:

rate / RUN / number of years / RUN / monthly repayment / RUN / original amount / RUN / balance

Example:

I bought a house seven years ago and took out a mortgage for £5500 at 11½% interest. My monthly repayment has been £70. I now want to sell my house and pay off the mortgage. How much will I have to pay?

Rate	1 1 · 5 RUN
Number of years	7 RUN
Monthly payment	7 0 RUN
Original amount	5 5 0 0 RUN

Balance = £3438

÷	G	00
#	3	01
1	1	02
0	0	03
.0	0	04
=	–	05
sto	2	06
+	E	07
#	3	08
1	1	09
=	–	10
ln	4	11
X	·	12
stop	0	13
=	–	14
▼	A	15
e^x	4	16
X	·	17
(	6	18
stop	0	19
X	·	20
#	3	21
1	1	22
2	2	23
÷	G	24
rcl	5	25
=	–	26
sto	2	27
–	F	28
+	E	29
stop	0	30
)	6	31
+	E	32
rcl	5	33
=	–	34
stop	0	35

CONVERSIONS

Metres to feet and inches

Execution:

metres / RUN / feet / RUN / inches

Note: This program may take some time to execute.

÷	G	00
#	3	01
·	A	02
3	3	03
0	0	04
4	4	05
8	8	06
–	F	07
(	6	08
–	F	09
#	3	10
1	1	11
=	–	12
▼	A	13
gin	1	14
2	2	15
1	1	16
▼	A	17
goto	2	18
0	0	19
9	9	20
+	E	21
#	3	22
1	1	23
=	–	24
sto	2	25
)	6	26
=	–	27
stop	0	28
rcl	5	29
X	·	30
#	3	31
1	1	32
2	2	33
=	–	34
stop	0	35

PERCENTAGE POINTS OF THE NORMAL DISTRIBUTION

Given any α with $0 < \alpha < 0{\cdot}5$, finds x to within about 2 sig. fig. so that the probability that a standard normal random variable exceeds x is α.

Execution:

α / RUN / x

For greater accuracy (·1% error) divide result by 1·006.

For still greater accuracy use execution sequence α / X / 1·0007 / RUN / ÷ / 1·006 / = / x

X	·	00
÷	G	01
=	–	02
ln	4	03
$\sqrt{x}$	1	04
sto	2	05
+	E	06
+	E	07
+	E	08
#	3	09
1	1	10
2	2	11
·	A	12
5	5	13
÷	G	14
(	6	15
rcl	5	16
+	E	17
#	3	18
7	7	19
X	·	20
rcl	5	21
+	E	22
#	3	23
5	5	24
=	–	25
)	6	26
–	F	27
+	E	28
rcl	5	29
=	–	30
stop	0	31
▼	A	32
goto	2	33
0	0	34
0	0	35

HYPERBOLIC FUNCTIONS

All the hyperbolic functions

Execution:

x / RUN / sinh x / RUN / cosech x / RUN / cosh x / RUN / sech x / RUN / tanh x / RUN / coth x /

Range:

$1{\cdot}0017 \times 10^{-4} \leqslant |x| \leqslant 7{\cdot}8566$

▼	A	00
e^x	4	01
+	E	02
#	3	03
1	1	04
÷	G	05
+	E	06
–	F	07
#	3	08
1	1	09
–	F	10
=	–	11
▼	A	12
arctan	9	13
+	E	14
=	–	15
sto	2	16
tan	9	17
stop	0	18
÷	G	19
=	–	20
stop	0	21
rcl	5	22
cos	8	23
÷	G	24
=	–	25
stop	0	26
÷	G	27
=	–	28
stop	0	29
rcl	5	30
sin	7	31
stop	0	32
÷	G	33
=	–	34
stop	0	35

Sample from Volume 2

QUADRATIC EQUATIONS

$ax^2 + bx + c = 0$

Roots x_1, x_2 if real

R ± iI if complex

Execution:

a / RUN / b / RUN / c / RUN /
- x_1 / RUN / x_2 / RUN / RUN / C/CE / C/CE / if roots are real
- I* / C/CE / RUN / R / if roots are complex

* error symbol displayed

After the sequence a / RUN / b / RUN / c / RUN / the display shows *either* (if the roots are real) the larger real root with no error indication *or* (if the roots are complex) the imaginary part and the error symbol. Continue with the appropriate execution sequence.

The error symbol will tell you whether the roots are complex. The sequence / RUN / RUN / C/CE / shown above after (x_2) is necessary before entering a new equation to be solved.

+	E	00
÷	G	01
–	F	02
X	·	03
sto	2	04
stop	0	05
=	–	06
▼	A	07
MEx	5	08
X	·	09
stop	0	10
+	E	11
+	E	12
(	6	13
rcl	5	14
X	·	15
)	6	16
+	E	17
▼	A	18
gin	1	19
3	3	20
2	2	21
$\sqrt{x}$	1	22
▼	A	23
MEx	5	24
–	F	25
stop	0	26
rcl	5	27
–	F	28
rcl	5	29
=	–	30
stop	0	31
$\sqrt{x}$	1	32
stop	0	33
rcl	5	34
stop	0	35

CIRCLES

Circumference and area:

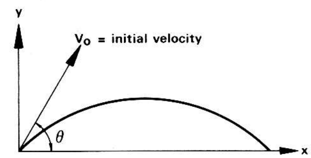
×	·	00
(	6	01
×	·	02
#	3	03
6	6	04
·	A	05
2	2	06
8	8	07
3	3	08
1	1	09
9	9	10
=	–	11
stop	0	12
)	6	13
÷	G	14
#	3	15
2	2	16
=	–	17
stop	0	18
▼	A	19
goto	2	20
0	0	21
0	0	22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Execution:

radius / RUN / circumference / RUN / area

PROJECTILES

Position relative to point of projection after time t



$x = v_o t \cos\theta$

$y = v_o t \sin\theta - \frac{gt^2}{2}$

Execution:

$\theta°$ / RUN / v_o / RUN / t / RUN / x / RUN / y

In S.I. units; g taken as $9{\cdot}81 ms^{-2}$.

▼	A	00
D→R	3	01
sto	2	02
tan	9	03
×	·	04
(	6	05
rcl	5	06
cos	8	07
×	·	08
stop	0	09
×	·	10
stop	0	11
sto	2	12
)	6	13
stop	0	14
–	F	15
(	6	16
rcl	5	17
×	·	18
×	·	19
#	3	20
4	4	21
·	A	22
9	9	23
0	0	24
5	5	25
=	–	26
)	6	27
=	–	28
stop	0	29
▼	A	30
goto	2	31
0	0	32
0	0	33
		34
		35

RELATIVITY

Fitzgerald contraction, time dilation and mass change.

$$T' = T\left(1 - \frac{v^2}{c^2}\right)^{½}$$

$$L' = L\left(1 - \frac{v^2}{c^2}\right)^{½}$$

$$M' = M\left(1 - \frac{v^2}{c^2}\right)^{-½}$$

Execution:

(i) v / RUN / c / RUN / T / X / RUN / T′

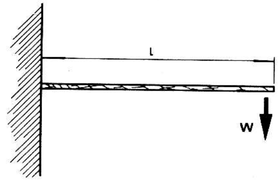
(ii) v / RUN / c / RUN / L / X / RUN / L′

(iii) v / RUN / c / RUN / M / ÷ / RUN / M′

÷	G	00
stop	0	01
X	·	02
–	F	03
+	E	04
#	3	05
1	1	06
=	–	07
$\sqrt{x}$	1	08
sto	2	09
stop	0	10
rcl	5	11
=	–	12
stop	0	13
▼	A	14
goto	2	15
0	0	16
0	0	17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

BEAM BENDING

Beam with one fixed end and load W at free end



$$\text{end slope} = \frac{W\ell^2}{2EI}$$

$$\text{end deflection} = \frac{W\ell^3}{3EI}$$

Execution:

ℓ / RUN / W / RUN / E / RUN / I / RUN / slope / RUN / deflection

sto	2	00
X	·	01
X	·	02
stop	0	03
÷	G	04
stop	0	05
÷	G	06
stop	0	07
÷	G	08
#	3	09
2	2	10
÷	G	11
stop	0	12
#	3	13
1	1	14
·	A	15
5	5	16
X	·	17
rcl	5	18
=	–	19
stop	0	20
▼	A	21
goto	2	22
0	0	23
0	0	24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

RESISTORS IN PARALLEL

(capacitors in series)
(inductors in parallel)
(conductors in series)

Pre-execution:

0 / ▲▼ / sto / C/CE / ▲▼ / ▲▼ / goto / 0 / 0 /

Execution:

R_1 / RUN / R_2 / RUN / $\frac{R_1 R_2}{R_1 + R_2}$ / R_3 / · · · / R_n / RUN / $R_{parallel}$

Alternative execution:

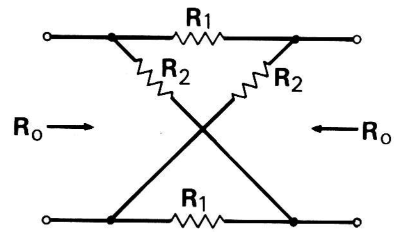
To find resistor R_2 required to make parallel combination of R_1 and R_2 = R:

R / RUN / R_1 / ▲▼ / ▲▼ / +/– / RUN / R_2

(R_1 must be greater than R)

÷	G	00
+	E	01
rcl	5	02
=	–	03
sto	2	04
÷	G	05
=	–	06
stop	0	07
▼	A	08
goto	2	09
0	0	10
0	0	11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

LATTICE ATTENUATOR SECTIONS



(must be balanced, constant impedance)

$a_v = a_i = a$ $A = -20 \log a$

Characteristic impedance = R_o

$R_1 = \frac{1-a}{1+a} R_o$ $R_2 = \frac{1+a}{1-a} R_o$

Execution:

either

/ ▲▼ / ▲▼ / goto / 1 / 3 / a / RUN / R_o / RUN / R_2 / RUN / R_1

or

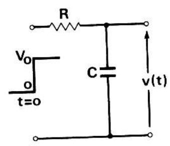
/ A / RUN / R_o / RUN / R_2 / RUN / R_1

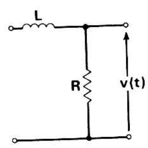
–	F	00
÷	G	01
#	3	02
8	8	03
·	A	04
6	6	05
8	8	06
5	5	07
8	8	08
9	9	09
=	–	10
▼	A	11
e^x	4	12
+	E	13
#	3	14
1	1	15
÷	G	16
(	6	17
–	F	18
#	3	19
2	2	20
–	F	21
)	6	22
X	·	23
sto	2	24
stop	0	25
=	–	26
stop	0	27
÷	G	28
(	6	29
rcl	5	30
X	·	31
)	6	32
=	–	33
stop	0	34
=	–	35

Sample from Volume 4

LINEAR CIRCUIT THEORY

Simple L–R or C–R circuit





$\tau = CR$ or $\tau = \frac{L}{R}$

Charge: $V_c(t) = V_o(1 - e^{-\frac{t}{\tau}})$

Discharge: $V_d(t) = V_o e^{-\frac{t}{\tau}}$

Pre-execution:

R / X / C / = / ▲▼ / sto / *or*

L / ÷ / R / = / ▲▼ / sto / *or*

τ / ▲▼ / sto / ▲▼ / ▲▼ / goto / 0 / 0 /

Execution:

t / RUN / V_o / RUN / $V_d(t)$

÷	G	00
rcl	5	01
–	F	02
=	–	03
▼	A	04
e^x	4	05
X	·	06
stop	0	07
=	–	08
stop	0	09
▼	A	10
goto	2	11
0	0	12
0	0	13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35