CENTRE FOR RESEARCH COLLECTIONS
EDINBURGH UNIVERSITY LIBRARY
GEORGE SQUARE
EDINBURGH
EH8 9LJ
TEL: +44 (0)131 650 8379
FAX: +44 (0)131 650 2922

BOOK-SCANNED 2011

TITLE: Ferranti Mercury Computer

N.B. Scanned as single pages.
Blank pages have been omitted.
Due to variations in the density of the text some words are illegible.
The text contains some pagination errors.
THIS IS THE BEST COPY AVAILABLE.

© Edinburgh University Library
This PDF is supplied for the purpose of research and private study.
No reproductions or further copies may be made without prior permission from Edinburgh University Library.

Please address all enquiries to Centre for Research Collections.
The Library wishes to be informed of work based on this Pdf copy, and would welcome a copy of any publication that makes use of it.

The University of Edinburgh is a charitable body, registered in Scotland, with registration number SC005336

FERRANTI
MERCURY COMPUTER

ANNOTATED
INPUT ROUTINE

A. GIBBONS.

FERRANTI
MERCURY COMPUTER

Input Routine

COMPUTER DEPARTMENT

Office and Works: WEST GORTON, MANCHESTER 12
London Computer Centre: 21 PORTLAND PLACE, LONDON W.1

LIST CS 240

JULY 1959

NOTES.

Programme is assembled sector by sector in page 1, and written to the drum when page 1 is full, or at the end of a chapter.

An item of information (e.g. instruction, number, integer, equation) is terminated by 'CR'. ',' (comma) has the same effect as CR except in a floating point number, when ',' introduces an exponent.

An item is compiled in sections, here called "phrases". A phrase is identified by its first character and terminated by the first character of the next phrase or CR. Characters which terminate the preceding phrase are:

v x n * (CR and usually) .

The function and b part of an instruction is automatically terminated after the third decimal digit, before any other character is read.

For example, the instruction

300 −2 x1 (5

is punched

300 Sp −2 x1 Sp (5 CR LF

and is processed:

(i) Read '300'
[Self-terminating]. Plant the binary equivalent of the '300' function in the programme.

(ii) Read 'Sp −2 x'
['x' terminates the phrase '−2']. Plant '−2' in the address part of the instruction.

(iii) Read '1 Sp ('
['(' terminates the phrase 'x1'] Add the value of x1 in to the address.

(iv) Read '5 CR'
['CR' terminates the phrase '(5'] Set label five.

The LF has no effect.

Items and Types.

The various different sorts of item have been given numbers, between 0 and 5. The number allotted to a particular sort of item depends mainly, though not entirely, on the number of registers occupied by the item. While an item is being read, the item number is kept in one of the working space registers of page 0.
The item number is used in the setting of labels (see below) and in the interpretation of the relative address *.

Associated with each function is a type number, between 0 and 7, which indicates how the address, as read from the tape, is to be processed. For example, the address in an accumulator instruction must be halved before being stored. Equations, ones, line and M directives, also have type numbers.

The tables of item and type numbers are on page 4.

July 1959

Labels and references.

When a phrase setting a label is read, a forty bit word (referred to as a 'label') is compiled, containing the routine number, the label number, the value of the label, and the item number of the item in which the label has been set. During the reading of a routine, the labels set are stored in a part of the computing store; at the end of the routine the labels are added to a list of all labels which is kept on the drum, and the part of the computing store is left clear for the labels of the next routine.

When a reference to a floating address is read, the input routine first looks to see if the relevant label is set; if so, the floating address is interpreted and added in immediately. If not, a forty bit word (referred to as a 'reference') is compiled, containing the routine number and the label number of the floating address to which reference is being made, the type number of the function in the item making the reference, and the address into which the floating address is to be added. This 'reference' is added to a list of unfilled references kept in a part of the computing store. At the end of each routine, this list of unfilled references is scanned, and those which refer to labels which have now been set are dealt with and removed from the list.

Details of the structure of labels and references appear on page 4.

Cues.

The references in cues (ACROSS and DOWN) are not filled in straight away, even if the symbolic address referred to has been set, because a cue requires details about the chapter to which it refers, and these details may not yet be known. (For details of cue structure, see the notes to the Chapter Changing Sequence on page 14). In the case of a cue (which is type 6) a reference is always compiled; when the reference list is scanned at the end of a routine those with type 6 are left alone, and filled in only at the end of a chapter.

Preset parameters.

Preset parameters are stored in exactly the same way as the labels of the current routine, but in a different part of the computing store. The structure of a preset parameter is identical to that of a label, even including a routine number, which is redundant in this case. The item number stored in a preset parameter is always 0 (i.e. 'label' set by an equation).
The preset parameter space is of course not cleared at the end of each routine.

The allocation of the computing store.

Page 0.	Constants and indicator.
Pages 1 – 15	Programme. The main body of the input, Chapter 1, occupies pages 1 – 14. Other chapters handle directives, etc. For these, sectors are brought down to some or all of pages 11 – 13 (the parts of Chapter 1 in these pages not being required for directives), and Chapter 1 is restored at the end of the directive action.
16.0 – 24.42	Unfilled references.
24.44 – 27.52	Preset parameters (x0 to x100)
27.54 – 30.62	Labels of the current routine (v0 to v100).
page 31	Working space.

The allocation of the drum.

Sectors 0 - 46	The Input Routine
Sectors 47 - 63	The Quickies.
Sectors 64 - 95	The label list
Sectors 96 - 111	The routine list
Sectors 112, 113	The chapter list
Sectors 114 - 126	Interlude storage space.
Sector 127	Working space.

The label list.

At the end of each routine, the labels of that routine which have been set are copied from the computing store to the label list on the drum; no spaces are left in the drum list for unset labels. Each label contains its own number and the number of its routine, and so can be found by a search of the relevant part of the label list. Finding the relevant part is one of the uses of:

The routine list.

There is one entry in the routine list for each V- routine, a 20 bit entry of which the first ten give the 'drum address', relative to 64.0, of the first label of the routine in the label list, and the second ten give the number of the chapter in which the routine appears. Sectors 96 - 111 contain 1024 20 bit words: if these are considered as numbered 0 - 1023, the entry for a routine is made in the appropriately numbered word, so that only words 1 - 999 are used. The entry is made when the routine heading is read.

The chapter list.

There is one entry in the chapter list for each chapter, a 20 bit entry of which the first ten give the first sector occupied by the chapter, the next five give the first page of the computing store, and the remaining five the last page occupied by the chapter. Entries are made in order of chapter number, so that only the 100 words 112.1 - 113.36 are used.

While a chapter is being read, its space in the chapter list is used only for storing its first page, which is stored in the last five bits. At the end of the chapter, indicated by a new C directive or an E or an M, the first page is shifted up five bits and the first sector and last page added at either side.

Chapter 1 of the input routine starts on sector 13. Sectors 0 - 5 contain the kick-off and selection of start procedures, binary routines, the Chapter changing Sequence, and the Error Print routine; sectors 6 - 11 contain the Post Mortems, and sector 12 the Fault Print and drum clearing routines.

— " —

Notation for jumps.

In the text, jumps are traced by straight arrows. Curved arrows are used to connect the settings of links to the locations jumped to when the links are obeyed. For example:

— " —

July 1959

ITEM AND TYPE TABLES.

STRUCTURE OF LABELS AND REFERENCES.

Item Table:

Item number	Item
0	Quicky
1	Floating point number.
2	Instruction
3	Integer
4	Equation or UP cue
5	ACROSS and DOWN cues.

Type Table.

Type number.	Equation symbol	Type	Adjustment to address.
0		57, 58, 99 functions, etc.	None.
1	=	Jumps, drum transfers, 62 function	None.
2	>	63 function.	Address doubled but no adjustment of most significant function bit.
3		1-, 3-, and 7- group functions 0-7.	Symbolic address may be halved or doubled if set by a label.
4	≠	40 bit functions	Address halved.
5		0- and 2- group functions 0-7	Address doubled with adjustment of most significant function bit.
6		cues	None
7		line and n directives	None.

A label is stored in forty bits as follows:

The first three ten bit words hold the routine number, label number, and the value of the 'flad' respectively. When the 'flad' is set by labelling an item, the 20 bit address is always stored, so that an underflow bit (for least significant half registers) and some overflow bits for addresses ≥ 16.0 are required. The fourth ten bit word contains these bits and the item number:

(most significant) item (least significant)

underflow overflow bits.

A reference is stored in forty bits as follows:

The first ten bits hold the routine number, and the first seven bits of the next half register hold the label number of the floating address to which reference is being made. The remaining three bits of the second half register hold the type number of the item in which the reference appears. The next ten bit word holds two indicators at the top, the first to say whether the reference is a v- or u- reference (0 if v, 1 if u) and the second, used only in cues, to indicate an "absolute address cue" (e.g. ACROSS 1.0/5) The last seven bits hold the half register within a sector, and the final ten bits hold the sector number of the location into which the symbolic address is to be added.

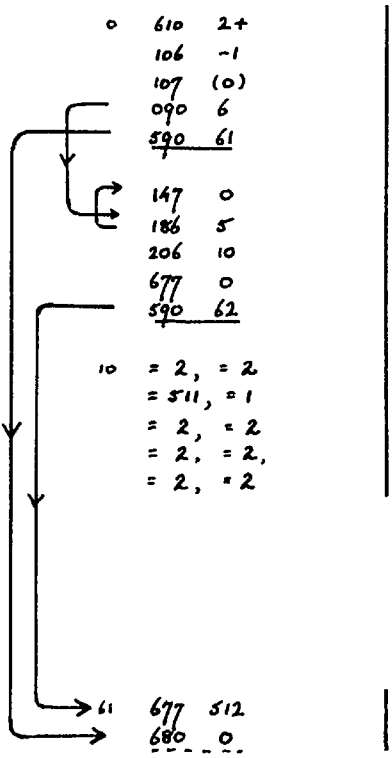
July 1959

SECTOR 0. -1.

Sector 0 is brought to page 0 and entered at line 0 when the Initial Transfer Button is pressed.

The "kick-off" in lines 0 - 14, 61 and 62, reads the initial handswitch setting, separates the handswitch 9 setting and for other settings puts n+1 into B6, where n is the number of the handswitch set.

It enters: engineers' routines if n = 1 (sector 511)
tele-input if n = 2 (sector 1)
and for other settings enters sector 2.



```
0   610  2+      read handswitches to 2+
    106  -1
    107  (0)     set B7 = handswitch setting (2^n)
    090  6       jump unless handswitch 9 set
    590  61      to enter sector set.

    147  0       } form n+1 in B6
    186  5       }
    206  10      table look-up for sector.
    677  0       select required sector.
    590  62      to enter the selected sector at line 63.
                 Sector table:
10  = 2, = 2     start with clear ; start without clear
    = 511, = 1   engineers' routines ; tele-input
    = 2, = 2     tele-output ; rescue
    = 2, = 2,        post mortems
    = 2, = 2     post mortem ; restart.

61  677  512     select sector set with handswitch 9.
    680  0       enter selected sector at line 63.
```

July 1959

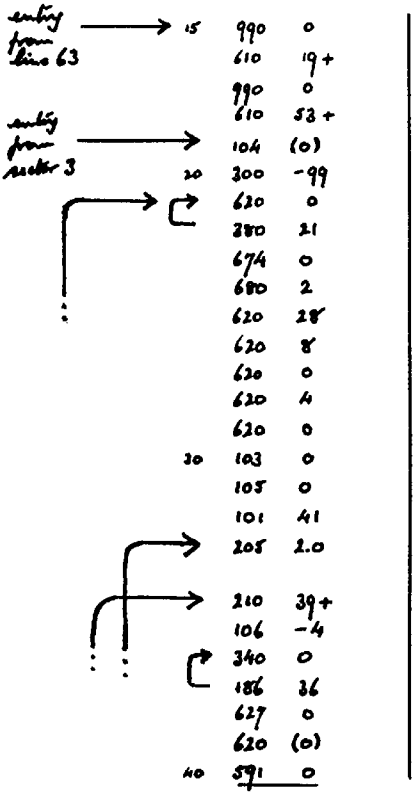
SECTOR 0 - 2.

Tele-output.

This is brought down to page 0 either by the kick-off (on handswitch 9 entry to sector 0) or by sector 3 (on handswitch 3 setting). The entry from handswitch 9 is to line 63, and in this case the first and last sector to be punched in binary are read from the handswitches, with a 99 stop before each handswitch read. When the sectors have been punched the routine comes to a further 99 stop before punching a loop stop warning character and one inch of erasor to end the tape.

On entry via sector 3, the first and last sector are specified by handswitch tapping and planted by sector 3 into the routine. At the end of the punching sector 3 is re-entered for another sector number to be specified by handswitch tapping: the 'first sector.' If this 'first sector' is non-zero, an automatic entry sequence is punched; if zero is tapped a loop stop warning character is punched as before.

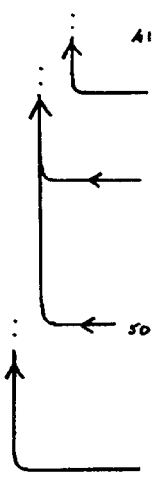
With each sector is punched a 10 bit check sum, which, added to the sum of the ten bit words of the sector, gives zero. Each sector is preceded by the sequence of characters 28, 8, 0, 4, 0 ("read 128 character pairs to 2.0 onwards") and followed by the sequence to write page 2 to the appropriate sector. One inch of blank tape separates sectors.



July 1959

SECTOR 0 – 3.

Tele-output (continued).



			after punching two characters :
	41	033 39+	subtract word from 33 for check sum.
		175 127	} cycle through page 2.
		185 33	
		303 0	check sum to Sac.
		101 47	set 81 for link after punching the check sum.
		590 34	to punch the check sum.
		620 26	"write page to sector" warning character
		304 0	sector number to Sac.
		101 51	set 81 for link after punching the sector number.
	50	590 34	
		620 2	page 2 to sector
		300 -9	set Sac for 1" blank tape
		174 (0)	compare 84 with last sector to be punched.
		184 21	back if more required
		990 10	stop (changed to a jump by sector 3)
		620 30	loop step warning character.
		300 -9	}
		620 31	} 1" erases at end
		380 58	}
	60	590 60	loop.
	63	590 15	handswitch 9 entry.

July 1959.

SECTOR 1. -1.

Tele-input.

A section on a binary tape consists of:

(i) a warning sequence, headed by a warning character
(ii) main information, including check sum (where relevant).

Between sections tele-input searches for a warning character and ignores all other characters.

The warning sequences are:

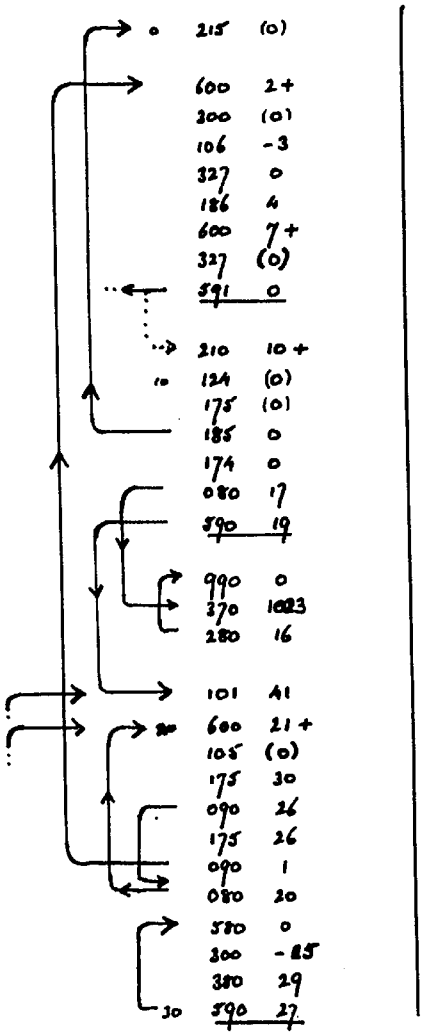
26, a, b, c	write page c to sector 32 a + b
27, a, b, c	read sector 32 a + b to page c.
28, a, b, c, d	read 32c + d character pairs from tape to the computing store from half-register 32 a + b onwards.
29, a, b	jump to line 32 a + b
30	host loop stop.

The check sum is such that the ten-bit sum of all the ten-bit words of the main information and the check sum together comes to zero. If the check sum is punched as 1023, no check is done.
Tele-input is entered by bringing sector 1 to page 0 and entering at line 63.

July 1959.

SECTOR 1. - 2.

Tape-input (Continued)



plant 10-bit word

B1 closed subroutine for forming a 10-bit word in Sac from two characters on tape.

main information : add into check sum in B4

test count ; come out before planting check sum

test check sum.

stop for check-sum failure
ignore check-sum failure if check-sum is 1023

entry to search for warning sequence.
character χ to B5
jump if $\chi = 30, 31$
to read 32a+b to Sac for $\chi = 26, 27, 28, 29$.
ignore $\chi < 26$ or $= 31$
host loop for $\chi = 30$.

July 1959.

SECTOR 1 -3.

Tele-input (Continued)

```
31  087  0        jump to 32a+b for χ = 29
                    χ = 28:
    210  0+       dump 32a+b in plant instruction.
    101  35       } to read 32c+d
    590  1        }
    330  1        } plant 32c+d-1 in compare instruction.
    210  11+
    104  0        set B4 for check sum
    105  -1       set B5 for plant modifier
    101  9        } to read 32c+d character pairs to half register
40  590  1        }   32a+b onwards.

    175  28       } from reading 32a+b : jump if χ = 28 or 29.
    090  31       }
                    χ = 26 or 27
    677  0        select sector 32a+b
    600  45+      } read c to 5ac
    300  (0)      }
    175  27       } jump if χ = 26
    080  50
    687  0        sector 32a+b to page c
    590  20       to search for next warning sequence.

50  697  0        χ = 26: page c to sector 32a+b
    106  -5       } last half register address of page c.
    327  0        }
    186  52
    327  127
    210  58+
    106  -127
    680  1        sector 32a+b to page 1                    } dumm
    206  (0)      } compare page c with page 1              } check
    276  1.63+
60  280  60       loop if different
    186  58
    590  20       to search for next warning sequence.

    590  19       entry.
```

July 1959.

SECTOR 2 - 1.

STANDARD PAGE 0; HANDSWITCH TAPPING; STARTING PROCEDURES.

This sector contains the standard instructions for page 0 while a programme is being obeyed. On entry to a programme this sector is left in page 0 and on sector 479, with lines 14 - 63 clear.
The sector is also entered at line 63 from sector 0 for all starting procedures except handswitches 1, 2 and 9.

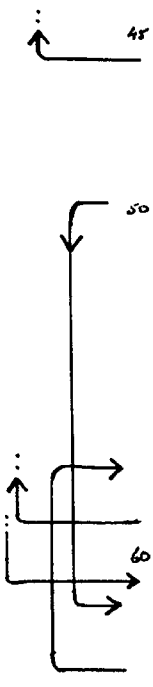
```
•    +0
     -1

     670  479
     690  0          } C.C.S. Entry
     670  478
     680  0
     597  0
     670  479
10   690  0          } Error Print Entry
     670  5
     680  0

     210  61+        store Sac (number tapped)
     206  40+        } select sector from table
     677  0
     176  4          } to enter selected sector if not tele-output.
     080  61
     670  3          } sector 3 to page 1 for tele-output.
     680  1
20   005  61+        BS = first sector tapped
     101  1.48       link for handswitch tapping for second sector tapped.

     300  0
     102  -2
     104  600        } wait for handswitches
     610  26+        } to be clear for about
     103  (0)        } 1/10 second
     184  24
     080  25
     610  30+        } digit tapped
30   103  (0)
     080  33
     590  29
     210  38+        } 10 x number tapped up to last digit - 1
     327  0
     327  (0)
     327  -1
     143  0          } add in last digit +1
     187  37         } three digits
     172  0
40   182  24         St = S
     370  0          out.
     591  0          sector table:
                     Rescue; post mortem instructions
                     Post mortem integers; post mortem numbers.

     = 9,  = 6
     = 6,  = 6
```

hand-switch tapping.

July 1959.

SECTOR 2 - 2.

Starting Procedures (Continued)



45	176	9	} after handswitch tapping: through if re-start
	080	13	
	677	0	} first sector to page 1
	680	1	
			tele-input automatic entry runs on into here with the first sector in page 1: select sector 3 and run on into it in line 63
	670	3	
50	590	62	
	670	0	from tapping "first sector" in tele-output: select sector 0
	101	1.6	} link and modifier for tele-output.
	105	-111	
	280	1.4	to sector 3 to punch automatic entry sequence if "first sector" is non-zero, and to punch the loop warning character if "first sector" is zero
	680	0	
	=0,	=0	spare
	101	45	link for handswitch tapping
	176	2	} to handswitch tapping except for "start"
	090	22	
60	670	12	to sector 12 for start.
	300	(≠ 1.2)	restore Sac or set Sac to ≠1.2
	680	0	
	590	57	entry from sector 0.

SECTOR 3. - 1.

PUNCH TITLE ; PUNCH SAC ; TELE-OUTPUT WITH TAPPING.

Brought to page 0 by sector 2 for title punching ; brought to page 1 by sector 2 for tele-output by tapping. The Punch Sac routine is B1 closed and is used in page 1 by the Error Print (Sector 5) ; it punches the integer n from Sac in the range 0 - 1023 without layout.

```
1.0   670   2
      680   0
      101   51
      590   22

      680   0
      210   1.10
      175   -105
      185   33
      590   56
      =0,   =27
1.10  (00),=59
      =2,   =0

      015   1.38+
      016   1.39+
      210   1.32+
      106   -30
      105   543
      370   0
      290   1.23
      330   1000
1.20  290   1.35
      320   500
      105   404
      125   177
      236   1.40+
      580   100
      290   1.23
      125   464
      090   1.27
      226   1.40+
1.30  126   10
      090   1.36
      370   (0)
      280   1.36
      590   1.16
      105   1
      625   0
      080   1.16
      105   (0)
      106   (0)
1.40  591   0
```

from tele-output after punching sector :
handswitch tapping back to page 0.
link for after tapping "first sector".
enter handswitch tapping.

from sector 2 if "first sector" ≠ 0 ; restore tele-output.
plant "first sector" in automatic entry sequence.
} loop to punch the automatic entry sequence.

to punch end of tape
} automatic entry sequence : "bring first sector to page 1 and sector 2 to page 0".

Punch Sac.
preserve B5
preserve B6
plant Sac for comparison
set B6 for count of 3
set B5 for start of "clutched count"
set 3' = 3
jump if $0 \leq n < 512$
3' = n - 1000
jump if $n \geq 1000$ to punch "1"
3' = n - 500
set B5 to start count at 3'
clutched count.
subtract 100, 10 or 1 from Sac.

cycle until Sac is negative
} adjust clutched count to give tape character in the bottom five bits of B5
add last 100, 10 or 1 back into Sac.
step B6
avoid suppression on last digit (B6 = 0)
} compare Sac with original value : no punching if Sac is unaltered (i.e. no digits yet)

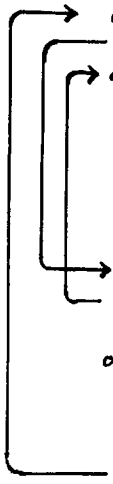
set B5 to punch 1 and B6 ≠ 0 for $n \geq 1000$
punch digit.
through after punching units digit.
} restore B5 and B6

out.

July 1959.

SECTOR 3 - 2.

Punch Title ; Tele-output with Tapping (Continued).



1.41	670	0	} from sector 2 after tapping first and last sectors: tele-output to page 0
	680	0	
	210	53+	plant last sector
	015	19+	plant first sector
	300	8	} change '990 1.0' in line 55 to '590 1.0'.
	210	55	
	590	19	enter tele-output
0.48	103	0	} Title punching : set B3 and enter loop.
	590	57	
0.50	203	63+	title character
	210	56+	
	106	-4	} top character
	340	0	
	186	53	
	627	0	bottom character
	620	(0)	
	073	1.61	} cycle until B3 = comm'.
	183	50	
	300	73	enter here on ENTRY : prepare to clear 13 lines.
0.60	400	1.62	entry cue to the accumulator
	670	4	} to sector 4 for store clearing.
	680	0	
	590	48	entry from sector 2.

THE CHAPTER CHANGING SEQUENCE. (C.C.S.)

During the execution of a programme the Chapter Changing Sequence is stored on sector 478, whither it is copied from sector 4 immediately before the programme is entered. The standard page 0 (sector 2) contains a sequence starting at line 4 which stores the current page 0 on sector 479, brings down sector 478 to page 0 and runs on into it at line 8.

After changing the chapter, the C.C.S. writes itself back to sector 478 (because some information about the chapter change may have to be preserved) and brings back either the standard page 0 from sector 479 or, if the new chapter starts in page 0, the first page of the chapter, and runs on into this at line 8. At this stage the entry point to the new chapter is in Sac, and the standard page 0 has a 597 0 instruction in line 8.

An ACROSS cue in the programme is translated into the two instructions:

300 4 *
590 4

followed by the four integers $s_1 - p_1$, p_1, p_2 and e

where the new chapter starts at sector s_1 on the drum and is to occupy pages p_1 to p_2 inclusive in the computing store; e is the address of the entry point to the chapter specified in the cue.

A dummy is inserted before the instructions if necessary to make the cue start at an even numbered register, so that the four short integers occupy one 40 bit word.

A DOWN cue is translated into the same piece of programme as an ACROSS cue, except that the ten-bit register containing p_1 also has the sign bit inserted as an indicator.

An UP cue is translated into two instructions:

300 1
590 4

and the first is not forced to be in an even register.

The 590 4 instruction in a cue simply enters the sequence in page 0 which calls for the C.C.S. In an ACROSS or DOWN cue, the address set in Sac by the 300 4 * is used to calculate the address of the information stored in the cue about the chapter required (i.e. the four short integers): 4 * is also the address of the register immediately following the cue, and in the case of a DOWN chapter change it is stored as the re-entry point for the appropriate UP.

The 1 set in Sac by an UP cue is used as an indicator that the chapter change is an UP; Sac = 1 is a setting which cannot arise from an ACROSS or DOWN cue.

From now on, for convenience, the four short integers $s_1 - p_1$, p_1, p_2, e, will be referred to as the cue.

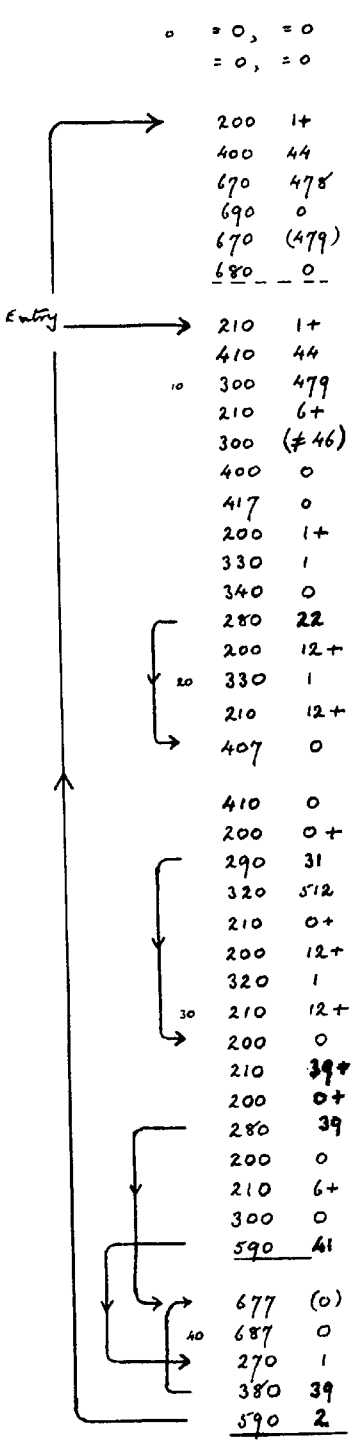
When the chapter changing sequence writes itself to sector 478 immediately before exit, it has the cue just obeyed in lines 0 and 1. On the next entry, then, this cue is still in lines 0 and 1, and this cue refers to the chapter from which the change is being made. The address 4 * is first planted on top of the old e, so that the cue becomes a re-entry cue to the present chapter, the cue which, if the present chapter change is a 'down', will be required for a subsequent 'up' change. This re-entry cue is now added to a list of cues the C.C.S. keeps within its own sector, starting at line 46. The C.C.S. also keeps an indicator of the next position in its cue list, known as the 'level indicator'.

On every entry to the C.C.S. the old cue from lines 0 and 1 is transferred to the next position in the cue list; however, only in a 'down' change is the level indicator advanced, so that in the other cases the old cue is never used, and is probably overwritten during a later chapter change. An 'up' change steps the level back one and takes its cue from the cue list; an 'across' or 'down' change takes its cue from where it has been set in the old chapter, using the address set in Sac to find it.

July 1959.

SECTOR 4 - 1.

THE CHAPTER CHANGING SEQUENCE.




```
0    = 0,   = 0
     = 0,   = 0

     200    1+
     400    44
     670    478
     690    0
     670    (479)
     680    0

     210    1+
     410    44
     300    479
     210    6+
     300    (≠46)
     400    0
     417    0
     200    1+
     330    1
     340    0
     280    22
     200    12+
     330    1
     210    12+
     407    0

     410    0
     200    0+
     290    31
     320    512
     210    0+
     200    12+
     320    1
     210    12+
     200    0
     210    39+
     200    0+
     280    39
     200    0
     210    6+
     300    0
     590    41

     677    (0)
     687    0
     270    1
     380    39
     590    2
```

} working space : an entry holds the cue referring to the chapter at present in the computing store.

entry point to Sac.
restore accumulator
} write up C.C.S. with cue list

} bring down sector 479 or first sector of chapter to page 0, and run on to line 8.

plant Sac (4 *) on top of entry point in old cue.
preserve the contents of the accumulator
} set the sector selected in line 6 to 479, in case the previous chapter used page 0 and so altered 6+.
pick up level indicator
} plant old cue, with new re-entry point, in the cue list.
4 * (Across or Down) or 1 (Up) to Sac.
} for Up, this leaves Sac = 0; for Across or Down, since Sac must be even on entry, it has the effect $s' = \frac{1}{2}s - 1$.
jump unless Up.

} decrease the level indicator by 1 and leave in Sac.

pick up new cue for Across or Down, or a cue from the list for Up.
plant in lines 0 and 1.
p_1 and Across/Down indicator.
jump if Across
} remove indicator bit in cue

} advance level indicator

$s_1 - p_1$ to Sac
plant in 'sector select' instruction
p_1 to Sac.
jump to bring down new chapter if $p_1 \neq 0$.
} if $p_1 = 0$, plant s_1 in 6+

$p_1 = 0$ to Sac.
enter "bring down chapter" loop

} bring down new chapter
(short register 1 contains p_2)

SECTOR 4 - 2.

STORE CLEARING; ENTRY.

```
44      :0, :0
        :0, :0
        410  0
        210  48
        (101) 0
        370  78
50      380  47
        670  2
        680  1
        400  44
        300  ≠ 1.16
        417  -2
        380  55
        670  479
        690  1
        300  -6
60      417  1.12
        380  60
        590  31

        590  46
```

} programme zero.

planet Entry case in C.C.S.

} clear B1 - B6

} "standard Page 0" to page 1.

programme zero to the accumulator

} clear lines 1.14 - 31.63

} write up "standard page 0" with lines 14 - 63 clear.

} clear lines 1.0 - 1.12

enter C.C.S.

entry from sector 3.

July 1959.

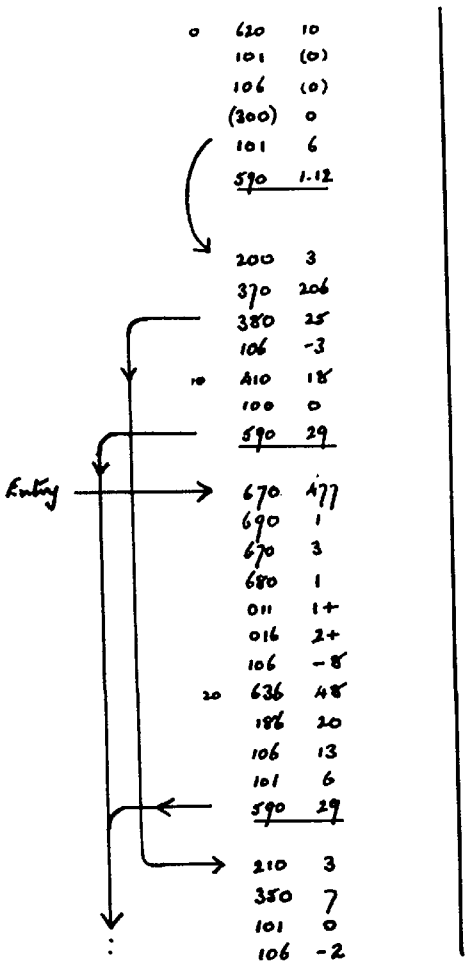
SECTOR 5. - 1.

ERROR PRINT.

Brought down to page 0 and entered at line 13 by the Error Print entry sequence in the Standard Page 0 (sector 2).

The routine prints the contents of 37 as the quicky number, the contents of 8 lines 1-6, (all in the range 0 - 1023), and the contents of the Accumulator as four integers.

It uses the 'Punch Sac' routine, on sector 3, in page 1, using sector 477 to preserve the original page 1. It finishes with a loop which hoots until handswitch 9 is set, when it restores the original pages 0 and 1 and enters page 1 at line 1.0



After punching B-line number
Punch =
} reset B1 and B6
}

contents of B-line to Sac (function planted from line 25)
set link
to punch contents of B-line.

After punching the contents of a B-line.
pick up '30b' function
test against '306' function
to punch next b if B6 not get punched.
set "A" modifier
plant accumulator
set B1 = 0
enter print sequence.

} preserve page 1 on sector 477

} punch Sac to page 1.
}

} preserve B1 and B6

} punch ERROR.
}

set "9" modifier
set link.
enter print sequence.

plant "30 b" function
b to Sac
set link for after punching B-line number
set "B" modifier.

rem on to print sequence.

July 1959.

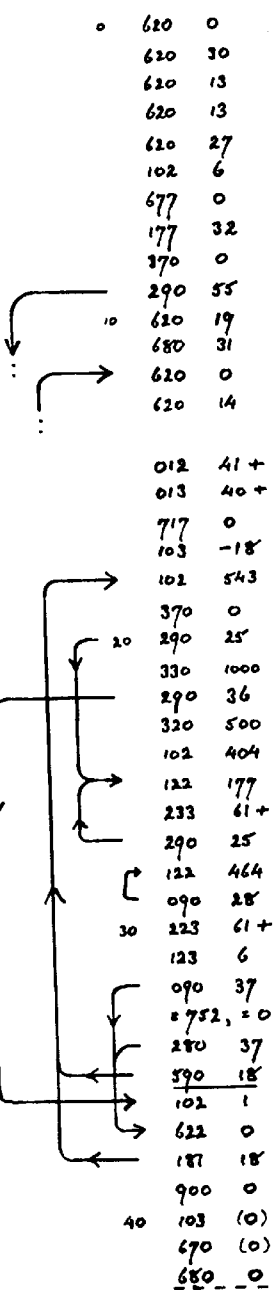
SECTOR 5. - 2

Error Print (Continued).

```
          620  30      punch CR
          620  13      punch LF
          620  27      punch letter shift
          626  4       punch letter
          620  0       punch figure shift
          080  1.12    to punch : quicky number after "Q"
                                  B-line number after "B"
                       and through after "A"
                        accumulator punching.
          620  10      punch =
          206  19+     part of the contents of the accumulator (B6=-3 first time)
          080  42      B1=0 first and last times (i.e. for exponent and most
          290  42                    significant part of the argument)
          620  11      } if Sac is negative, punch "-" and negate.
          360  -1      }
          320  1       }
          101  49      set link.
          590  1.12    enter "punch Sac".

          = 30, = 13   } table for punching
          = 27, = 5    }      "ERROR".
          = 18, = 18   }
          = 15, = 18   }
          = 0,  = 0    }

          176  0       after punching part of the accumulator
          620  15      } punch "," and cycle on B6
          186  36      }
          001  1+      } restore B1, B6
          006  2+      }
          670  477     } restore page 1.
          680  1       }
          580  0       } hoot while handswitch 9 is not set.
          300  -20     }
          380  58      }
          610  60+     }
          100  (0)     }
          090  56      } restore page 0
          670  479     }    and run on to line 1.0
          680  0       }
```

POST MORTEMS.

SECTOR 6 - 1.



```
0   620  0
    620  30
    620  13
    620  13
    620  27
    102  6
    677  0
    177  32
    370  0
    290  55
10  620  19
    680  31
    620  0
    620  14

    012  41 +
    013  40 +
    717  0
    103  -18
    102  543
    370  0
20  290  25
    330  1000
    290  36
    320  500
    102  404
    122  177
    233  61 +
    290  25
    122  464
    090  28
30  223  61 +
    123  6
    090  37
    =752, =0
    280  37
    590  18
    102  1
    622  0
    187  18
    900  0
40  103  (0)
    670  (0)
    680  0
```

from line 63,
punch ØCR LF LF λ

set B2 to return to this sector
select required sector
Bt = S - 32
St' = S
jump if S < 512, through for sector ≥ 512
punch 'S'
sector to page 31
punch Ø
punch Sp.

plant return sector from B2

punch Sac routine.
Alters B1, B2. Counts in B1 the number of digits punched.
Preserves Sac in the exponent of the accumulator.

Used for sector or page number, for integers, line parts of addresses, and exponents.

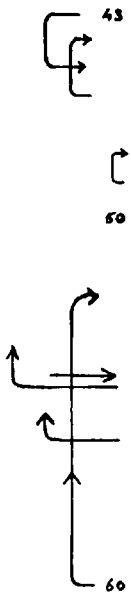
instruction St' = S - Exp.

restore Sac from the accumulator exponent
restore B3
select sector, planted in line 14.
bring down to page 0 and enter at line 43.

July 1959.

Post Mortems (Continued)

Sector 6 - 2.



43	290	45
	300	-1
	370	32
	290	44
	105	-4
	327	0
	185	48
50	704	0
	210	52+
	103	(100)
	676	1
	680	0
	090	10
	620	16
	590	12
	303	1
	157	31
60	183	53
	990	0
	304	1
	590	0

entry by rm on from the same sector, after punching sector or page number.
S' = S for $0 \leq S \leq 31$
S' = -1 for $S \geqslant 32$

Sac = first 40-bit address of page (for P)
or 31.0 (for S)
leaves B5 = 1 for entry to sector

sector or page number to B4

B3 = first 40-bit address for punching.
address part used by the punch Sac routine.

enter : sector 7 for instructions
sector 8 for integers
sector 9 for numbers.

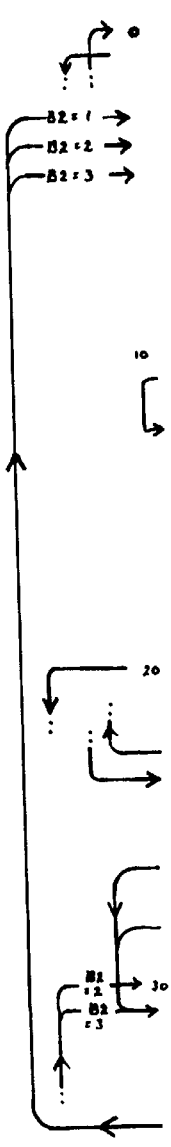
from line 9 : back if Sac $\geqslant 32$
punch 'P' for $0 \leq S \leq 31$

entry after punching a complete word :
Sac = next address
test for end of a page, advance B3

stop at the end of a page
Sac = next sector or page number.
entry or return.

Post Portems (Continued)

Sector 7 - 1.



```
0    101  512
     590  24
B2=1 010  3+
B2=2 126  (0)
B2=3 156  63
     102  -1
     330  10
     122  1
     290  6
     717  0
10   670  8
     185  13
     102  31
     680  0
     627  0
     620  14
     410  44
     205  2.44+
     106  -5
     172  1
20   080  49
     210  25+
     350  1
     280  0
     205  2.44
     107  (0)
     147  0
     290  31
     127  512
     590  31
B1=2 126  1
B2=3 210  3+
     340  0
     186  32
     006  3+
     592  1
```

} set B1 for + after address

} double B6 if B2=2, not if B2=1 or 3;
remove top bit to leave line in B6.

} first decimal digit of page number to B2,
second digit -10 in Sac.

store Sac in accumulator exponent.
select sector 8
} for suppression of first page digit:
step count in B5
run on to sector 8 in line 14

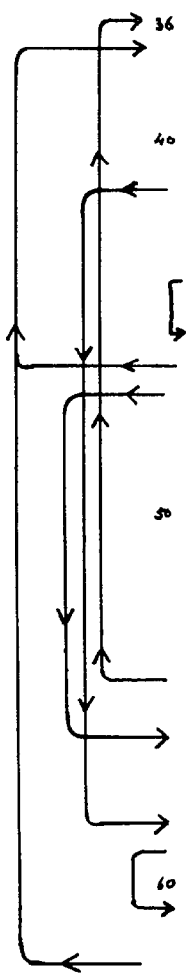
from sector 8 : punch B-digit.
punch 'Sp.'
dump instruction pair.
address part of instruction to Sac.
set B6 for shift down for page number.
} jump unless half-register type.

preserve address
} jump if odd address
(B1=1 at this stage)
top function bit to St.
restore address to Sac.
halve address

} adjust address for first or second
quarter of the store.

B6 = -4 for long register type
store address
} shift down 5 or 6 places for
page number.
address to B6
to form line in B6

Sector 7 - 2.



```
36  105  768   from entry for new word : set count in B5
    403    0   word to accumulator
    410   44   } function and 6 digits
    205  2.44  }      to sac
40  340    0   } to shift off 6 digits.
    340    0   }
    590   58

    171    0   } entry after punching an address:
    090   46   } punch '+' if B1 is negative.
    620   26
    175  771   } test count.
    185   37
    590   56   to return to sector 6 for new word.

    092   28   from line 20 : back unless B2=0 (i.e. address
                 is an integer)
50  102    7   set B2 to return to this sector
    106    6   re-set B6
    670    6   select sector 6 .
    620   14   punch SP.
    185   13   to enter sac punch. Step B5

    590   36   entry for new instruction pair.

    670    6   } exit after end of a word.
    680    0

    340  632   from line 42: Sac = function bits = 632
    290   61   jump if function < 120
60  330   64   remove top function bit for functions with value 120-127
    670   11   to sector 11 for function table look up.
    680    0
    590   37   re-entry after punching first instruction as two integers.
```

Sector 11.

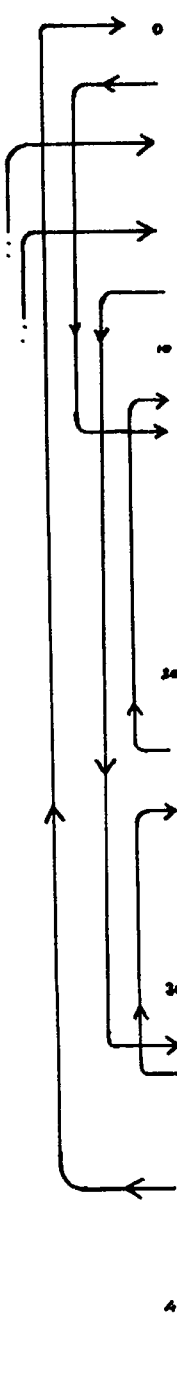
0-59 : Function Table : One ten-bit word per function.
Top two bits : address type indicator : 0 : Integer
1 : 10-bit address
2 : 40-bit address
3 : 20-bit address
Next four bits : second decimal function digit
Bottom four bits : first decimal function digit
An unassigned function has a zero entry.

```
60  207  -196   pick up entry into Sac.
    670    8   } return to sector 8
    680    0
    590   60   entry.
```

July 1959.

Post Routines (continued).

Sector 8 - 1.

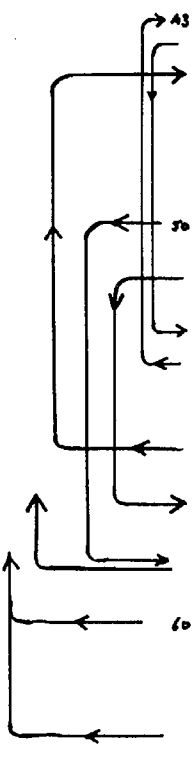


```
0   207  38     6 digit to Sac (punching character)
    670   7     } return to sector 7
    590  13
                 } new line: punch CR LF.
    620  30
    620  13
    403   0     saved to accumulator
    102   8     set B2 to return to this sector.
    410  44     plant word
    280  32     through for integer or unassigned function.
    205  2.44   integer to Sac.
10  620  10     punch =
    101  -5     set B1 for key-out count.
    670   6     } to punch integer.
    680   0

    202  38     } from sector 7 : punch first
    627   0       digit of page number.
    900   0     } punch second digit of page number.
    207  43     }
    627   0
    620  28     punch ". "
20  306   0     line number to Sac.
    102   7     set B2 to return to sector 7
    106   6     re-set B6
    590  12     to sector 6 to punch line number.

    350  15     collate out decimal function digit.
    207  38     } punch function digit.
    627   0     }
    300  (0)    restore Sac.
    101  -3     } shift down 4 places.
    340   0       Leave B1=1.
20  181  29     count in B2
    132   6     preserve Sac.
    210  27+    through punching loop twice.
    090  24     indicator bits to B2
    002  27+    function and 6-digits to Sac.
    205  2.44   6-digit to Sac.
    350   7
    590   0

    =16 , =1   } digit punching
    =2  , =19  }   table.
40  =4  , =21  }
    =22 , =7   }
    =8  , =25  }
```

July 1959.

Post Mortems (Continued).

Sector 8 - 2.



→ 43	171	0	} from punching integer or exponent : spaces for lay-out according to the count in B1.
	181	53	
	305	1	Sac : next integer count.
	290	51	jump after exponent punching.
	350	1	St = parity of count
	107	-1	prepare to set St = 0
	175	771	} test count.
30	185	58	
	670	6	} to return to sector 6 for new word.
	590	57	
	620	14	} lay-out spaces.
	590	43	
	105	767	} initial entry to punch new word as integers
	590	45	
	680	0	return to sector 6
	380	6	jump if count has odd parity (not new line). St' = 0
	176	6	} to punch integers on a new line if not in instruction punch.
60	080	3	
	670	7	} back to punch second half of word as an instruction.
	680	0	
	590	3	entry from function table look-up.

SECTOR 9. -1.

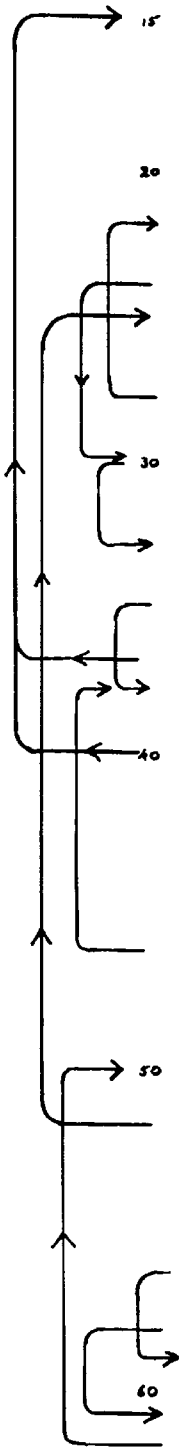
RESCUE.

```
0   106   -30
    677     1      } write up page 1-31
    696    31
    320     1
    186     1
    670   479
    680    31      } write up sector 479 on page 0
    677   -31
    690    31
    677     0      } restore page 31
10  680    31
    580     0
    300   -30      } boot.
    380    13
    590    11
```

July 1959.

Post Mortems (Continued).

Sector 9 - 2.

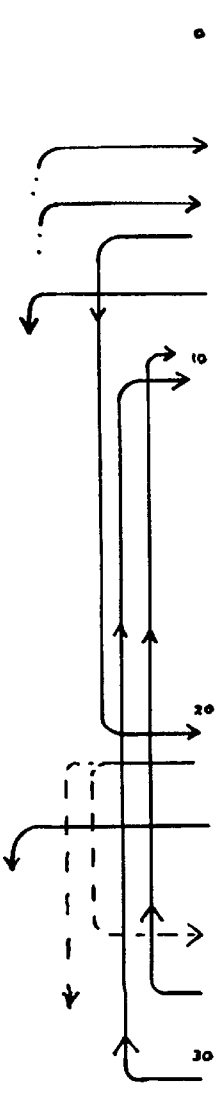


```
15  680  0        exit sector 10
    +0
    -1
20  +1000000
    620  3        punch * for unstandardised number.
    900  256      reset SC
    590  30
    101  3        set B1 for "sp".
    440  16       standardise the accumulator
    777  256      compare the new exponent with the old.
    080  22       jump if the number was unstandardised.
    620  14       "sp" for standardised number.
30  490  33       } if the accumulator is negative, multiply by -1
    520  18           and set B1 for "-" punching.
    101  0
    621  11       punch "sp" or "-"
    107  -78      set B7 = -78 for exponent count.
    280  38       jump if number exponent ≠ -256
    400  16       clear accumulator for exponent -256
    590  15           and enter sector 10.
    410  60       store accumulator
    700  -1       } to sector 10 when the exponent of the
40  090  15           accumulator ≥ 1
    330  6        decrease decimal exponent count by 6.
    540  20
    410  62       } number × 10^6
    400  60
    520  20
    440  62
    590  38       back to test exponent against 1.
    =0,  =0      } spare
    =0,  =0
50  670  10       select sector 10
    900  256      } jump if the exponent of the accumulator is in
    290  25           the standard range -256 to +255
    670  8        } to line 55 of sector 8
    680  0            to punch the number as four integers.
                        Entry from sector 6:
    175  512      } if B5 is -ve, the last number was punched as
    090  59           four integers; jump to punch CR LF.
    350  3
    280  61       } CR LF every four numbers
    620  30
60  620  13
    403  0        next number to the accumulator.
    590  50       to test the exponent.

    590  0        rescue entry.
```

November 1959.

Post Mortems (Continued).

Sector 10 - 1.

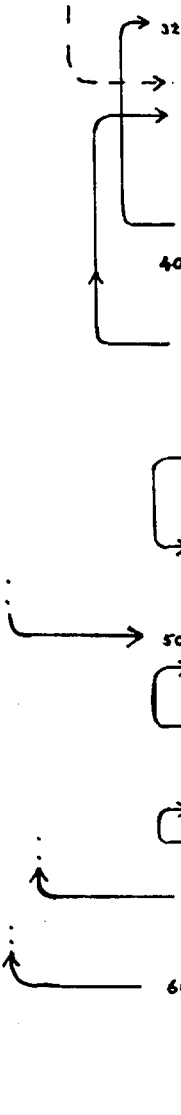


```
0   = -3,  = 410                    +0·1
    = 614, = 409

      + 10

    625  0                          punch digit
    101  -2                         reset B1 to second value.
    172  -9                         } jump except first time;
    182  21                             skip B2.
    620  28                         punch "." after first digit.
    590  50

10  620  26                         punch "+" for positive exponent
    105  3                          set B5 positive to indicate exponent punching (see sector 8).
    102  8                          set sector 8 for return from punching exponent.
    680  0                          to sector 6 to punch exponent.

    = -24, = 667                    1/2 × 10^-7
    = 508, = 429

                                    Entry from sector 9:
    102  13                         set B2 to adjust power of 10 in steps of 10^12.
    101  6                          set B1 to initial value.
    670  6                          select sector 6
    410  18                         store the accumulator
20  400  0                          0·1 to the accumulator for generating 1.
    520  2                          accumulator × 10
    091  28                         to line 34 first time; to line 26 after punching
                                     the argument (when B2=0).
    590  50                         to form the next digit of the argument.

    +1,000,000,000,000              + 10^12

    320  76                         Sac = decimal exponent.
    620  15                         punch "3" before exponent.
    290  10                         jump if the exponent is positive.
    620  11                         } punch "-" and negate Sac
30  167  -1                             for negative exponent.
    380  11
```

November 1959.

Post Mortems (Continued).

Sector 10 - 2.



	32	400	16
		522	-2
		410	16
		322	-1
		290	41
		532	-2
		440	18
		490	32
	40	400	16
		332	-2
		132	11
		090	35
		520	0
		410	0
		280	48
		400	14
		520	14
		440	18
	50	105	543
		125	177
		481	4
		490	51
		441	4
		125	464
		090	55
		175	906
		180	4
		620	1
	60	187	6
		= 0 ,	= 0
		= 0 ,	= 0
		= 0 ,	= 0

last 10^v

} increase to current 10^v.

Sac = v + 12 - 78 or v + 1 - 77
jump if about to form 10^{84} or 10^{77}
- next 10^v

} test number against next 10^v; jump back if the number is bigger than the new 10^v.

current 10^v

Sac = v - 77 (first time) or v - 76 (second time).

} step B2; back if B2 = 2 to adjust v in steps of 1; though with B2 = -9 when the value of 10^v immediately below the number has been formed.

} store 10^{v-1}

jump unless v = 76
if v = 76, prepare to round with $\frac{1}{2} \times 10^{-14}$ (which has no effect on a number $\geq 10^{76}$)
$\frac{1}{2} \times 10^{v-8}$ as rounding constant
add to number.

} form the tape code for the next decimal digit, using the clutched count,
and 10^v first time : register 16, B1 = 6
10^{v-1} subsequently : register 0, B1 = -2.

re-add last power of 10 subtracted

} complete the clutched count

} jump unless the first digit is "10" - "10" can only occur the first time due to rounding. Set Bt ≠ 0.

punch 1
jump to punch "." without altering B1;
increase decimal exponent in Sac by 1.

} spare.

Sector 11 : See page 23.

SECTOR 12 -1.

FAULT PRINTING.

This is entered if during input of a programme a fault is detected on the programme tape (such as an impermissible character). It is brought to page 0 and entered at line 13 from sector 38 (q.v.), which is normally in page 0 during programme input. The original page 0 is stored on sector 127, whence it is brought back on exit from the fault print routine.

The fault printing is entered with the fault number in Sac. After the printing the routine comes to a perpetrable stop, after which characters on the input tape are read and ignored (apart from being copied to the punch) until CR is encountered, when the original page 0 is restored and the main input is re-entered.

C 13 T 0
R 99

```
0     = 0 (10,   = 30
      = 13,      = 30
      = 27,      = 6
      = 1,       = 21
      = 12,      = 20
      = 0,       = 14 (12

      626  0   86
      600  0+v7
      106  (0)  (7
      176  30
10    080  v16
      300  v/29
      680  0

fault print entry.
      210  0+v5
      106  v10 ∞ 12
      636  v12      (3
      186  v3
      200  0+v5
      670  35
      680  12
20    101  v5
      590  12.0
      = 0,  = 0     (8
      = 0,  = 0
      300  (0)      (5
      670  127
      680  15
      006  14 v/1
      620  14
      990  0

30    370  9
      280  1v7

      010  15.57
      690  15
      300  1v/53
      590  12
```

table for punching "FAULT".

punch character read.
read next character.
to 86
} back to copy to punch unless CR read.
set Sac to v/29 to restore Chapter 1.
bring down and enter original 'page 0'

preserve Sac, i.e. fault number.
} punch "FAULT"

} "punch Sac" routine (R 47) to page 12.
set return address.
punch fault number.
} programme zero (used for store clearing)

restore fault number to Sac.
} 'current page 0' to page 15.
last character read to 86
punch 'f'.
stop.

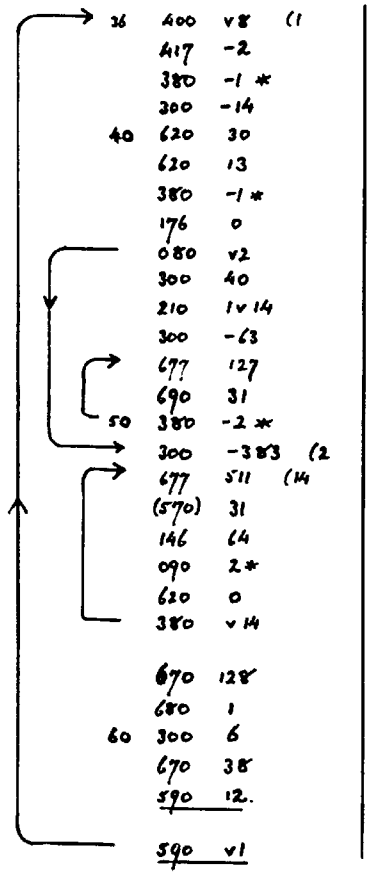
} to read to CR if not fault 9.

} if fault 9, clear 'predicted last page' setting in the current page 0.
link to return to Chapter 2.
to re-enter page 0.

SECTOR 12 - 2.

START WITH OR WITHOUT CLEAR.

Brought to page 0 and entered at line 63 from sector 2. Entered with Sac set to 33 (long address of 1. 2) and B6 = 0 for start with clear and = 1 for start without clear.



```
36  400  v8  (1
    417  -2
    380  -1 *
    300  -14
40  620  30
    620  13
    380  -1 *
    176  0
    080  v2
    300  40
    210  1v14
    300  -63
    677  127
    690  31
50  380  -2 *
    300  -383  (2
    677  511  (14
    (570)  31
    146  64
    090  2*
    620  0
    380  v14

    670  128
    680  1
60  300  6
    670  38
    590  12.

    590  v1
```

programme zero.
clear ff 1 - 31.
punch CR and 15 LFs.
jump if no clear.
plant '690' function over dummy.
clear sectors 64 - 127 (lists, etc)
clear sectors 128 - 511 (if start with clear) and punch one row of blank tape every 4 sectors, to give approximately 10 inches of blank tape.
sector 128 to page 1.
set Sac to entry point for bringing down chapter 1.
enter main input.

July 1959

CHAPTER 1.

(Section 13 - 25).

Brought to pages 2 - 14 and entered at v1/6 by R 102 (sector 38) which is in page 0.

C1 P 2 - 14

R 40
Fault 13

fault 13 detected →

```
        + 101
300  13  (1
590   9
```

} to punch 'fault 13'.

R 7
Build exact number in the Accumulator.

first decimal digit →
subsequent decimal digits →

```
400  0
103  v1
520  16  (1
440  14
590  v/1
```

first digit: clear the accumulator. reset B3 for subsequent digits.
} acc × 10 + new digit
return to read.

R 8
Halve or double the contents of the accumulator according to function type.

entry →
entry from R 27 →

```
300  0
207  v1  (3
727  0
717  0
591  0
+) = 29,  = 0
  = 0 (1,  = 0
  = 1,  = 0
  = -1,  = 1
  = 0,  = 0
```

function type to Sac. Set by R 9, R 13, ends, M, L.
required increment to exponent
} add increment to exponent.

out.
see R 27.

} table: double for types 2, 5
halve for type 4.

R 6
CR,), ϕ and LF entries. Start item.

CR →
) →
ϕ, LF →
read new item →

```
104  v1
592  0

172  v/37  (2
080  v/19
033  0 + v3/39
173  v5/39
080  v/19  (3

103  v/9  (1
010  0 + v/84
010  0 + v/18
400  0
410  50
102  v/1
```

} terminate last phrase.

} fault 3 on) unless in decimal input, i.e. after + or -
subtract decimal point switch from B3.
} fault 3 if digit or decimal point read before).
Also fault 3 on ϕ or LF except at the beginning of an item.
set B3 so that a decimal digit starts a fraction
} reset sign switch and equation switch.

} clear accumulator and address build working space.

set B2 so that a terminating character returns to read (only CR or , in this case) and runs on to read.

↓

R 1
Read character χ

Sf. Ev. →	600	1+v	} χ to B6
	106	(0)	table look up according to χ
	206	v/15	set Bt=0 if at the start of an item.
	172	v/1	
	290	v2	} jump according to B7 if χ is not a decimal digit (i.e. to address in table in R 15) or according to B3 if decimal digit. If the latter case, the table look up has sent the value of the digit in Sac, and this has been planted in 15+ for conversion to floating point.
	597	0	
	370	10 (2	
	297	0	
	210	15+	
	593	0	
λ →	670	46 (1	} letter shift table and R 81 to page 15.
	680	15	
	590	v/81	enter R 81.

R 3
n and v entries.

n →	330	511	} Sac = n/v switch : 0 if v −512 if n
v →	330	1v	
	210	0+v2	presence switch
	101	v1	set link for return after test if O.K.
	290	v/24	jump to test if O.K. if v.
	172	v/1	} n : to test if O.K. except at the beginning of an item (i.e. except for query n).
	080	v/25	
	670	37	} query n : bring R 71 to page 11 and enter.
	680	11	
	590	v/71	
	104	v2 (1	set B4 for return after terminating previous phrase.
	592	0	terminate previous phrase : STANDARD "592 0".
	300	(0) (2	pick up n/v indicator or abs. address indicator (see 2v4/17)
	210	0+v1/23	plant in R 23.
	102	v/23	set B2 to terminate v
standard 'read exact number' entry. →	400	0 (3	clear accumulator
	103	v/7	set B3 for first digit of exact number.
	590	v/1	return to read

R 80

L, P or S on right hand side of an equation.

```
181   v1
590   v/19
271   2v5/81   (1
280   v
171   511
090   v9/81
015   0+v4
104   v2
592   0

006   0+v1/31  (2
026   47
406   0
410   44
102   v3
590   v3/3

173   v/7      (3
080   v/19
105   (0)      (4
205   63

175   -3
080   v5
210   25+
200   62
210   27
400   26
440   24.
590   21 v/34
210   23       (5
400   22
590   21 v/34
```

to R 81. ←

compare B2 setting (transferred to Sac) with entries in table: fault 8 if no entry found to correspond.

back to R 81 if B1 ≤ −2, i.e. if S or P in C directive.
plant letter indicator (L, P or S)

terminate previous phrase

left hand side v or x switch
plus v or x number
appropriate label or parameter to line 44
return to read rest of phrase. (CR expected).

fault 3 if decimal digit read.

letter indicator
pick up to Sac: h if L
μ if P
ι if S

jump unless L.

½h + 64p = current address L to accumulator in floating point form

enter R 34 to add in to label.

p or s in floating point form to the accumulator
enter R 34 to add in to label.

R 10

W, → entries.

```
106   750
300   -16   (1
380   *
(580) 0     (2
186   v1
300   648
260   v2
210   v2
610   1+ *
106   (0)   (4
166   (0)   (3
090   -1v1
006   0+v4
016   0+v3
090   v/1
590   v
```

W, →

boot or dummy.

change boot or dummy instruction.

read handswitches.

back if top switch not changed.

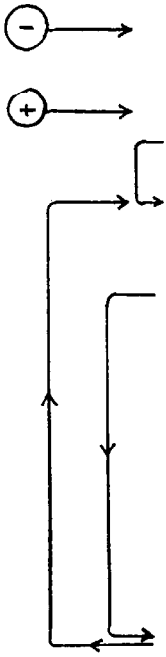
out if switch changed to zero

back if switch changed to non-zero.

July 1959.

R 12

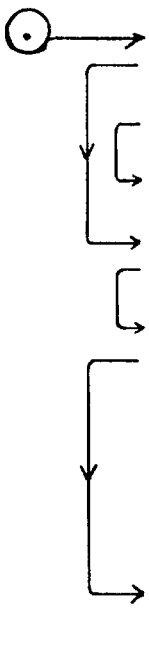
+ and − Entries.



```
300   -1                    } set - switch
210   0+v/84
300   2v                    } Sac = v6 for +
320   -3 n v6                     -1 v 6 for -
080   v1                    } to decimal input
590   v/39                    if at the beginning of a line.
210   1+ *   (1             } test B2. (Test for after page setting
072   (v6)                        only for - entry).
370   v7
080   v2                    } exit to R 56 if P
177   512 v6
090   v2/56
173   v1/7                  } if decimal digit not read, exit to test if
080   4v1/13                      B3 is set to v/7
176   26                    } fault 3 if terminal -
080   v/19
290   v/19                    terminal + not allowed in exponent.
440   20                      add ½ for terminal +
103   v/19                    fault 3 if decimal digit now read.
590   v/1                     return to read.

380   v1     (2             } fault 3 if B2 setting not
590   v/19                    found in table.

= v/57 , = v/32   (6        P   ; abs. address, integer, or exec address

= v2/64, = v1/49            M   ;   L

= v/31 , = v2/38  (7        right hand side of equation ; exponent of f.p. number.
```

R 14

• Entry.



```
172   v2/50                 } jump unless • in routine number
080   v3
450   30                    } fault 3 unless routine number already read
490   v4                          is ≥ 1000
590   v/19
440   30     (4             } restore number and return to read.
590   v/1
173   v/7    (3             } ignore • if no decimal digit read
080   v1                          in exact number.
590   v/1
173   v1/7   (1             } jump if not during
080   v2                          exact number.
707   6                     } acc. x 64 to L 50

717   0
416   50
400   0      (5             clear accumulator
590   v/1                   return to read.

172   v/37   (2             } fault 3 unless in floating point input.
080   v/19
590   v4/39                 to set decimal point switch.
```

July 1959.

R 15

Figure Shift Table.

= v3/6 ,	= 1	φ	;	1
= 2 ,	= v/35	2	;	*
= 4 ,	= v/5	4	;	(
= v2/6 ,	= 7	)	;	7
= 8 ,	= 2v/13	8	;	≠
= 1v/13 ,	= v/12	=	;	−
= 1v/3 ,	= v3/6	v	;	LF
= v/1 ,	= v/30	sp	;	,
= 0 ,	= v/13	0	;	>
= 1v/80 ,	= 3	≥	;	3
= v/10 ,	= 5	→	;	5
= 6 ,	= v5/17	6	;	/
= v/2 ,	= 9	×	;	9
= 3v/12 ,	= v1/1	+	;	λ
= v/14 ,	= v/3	.	;	n
= v/6 ,	= v/1	CR	;	Er

R 18

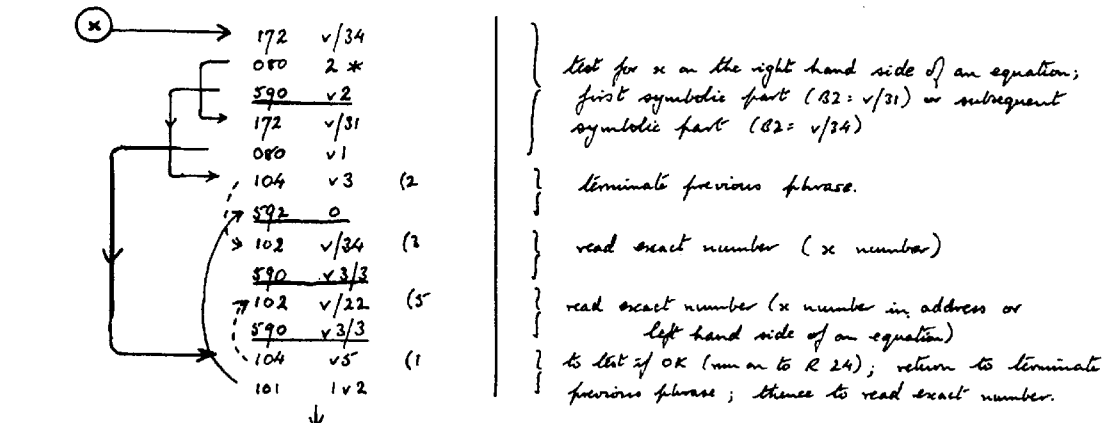
Overflow (Enter with the overflow in Sac).

100	(0)		Set B6 according to indicator: −1 if equation (R 31) 0 if address (R 9) (this indicator is also 1 if integer (R 13). used by R 32).
080	v4		jump unless address
006	0+~~v8~~		type to B6
176	5		jump unless type 5
080	v4		
370	2		fault 2 if the overflow is ≥ 2 or < −2
290	v1		
370	−2		
290	v2		
300	2	(1	
590	9		
350	1	(2	single overflow bit only
280	v3		out if overflow is −2 or 0.
591	0		
006	61+	(3	4
206	63+		function part of instruction to Sac.
320	512		change m.s. digit.
216	63+		replace.
591	0		out.
			types 0 − 4:
370	0	(4	out if no overflow
280	v5		
591	0		
370	−1	(5	fault 2 if overflow ≠ −1
280	v1		
591	0		out if overflow −1 (i.e. treat address modulo 1024).

July 1959.

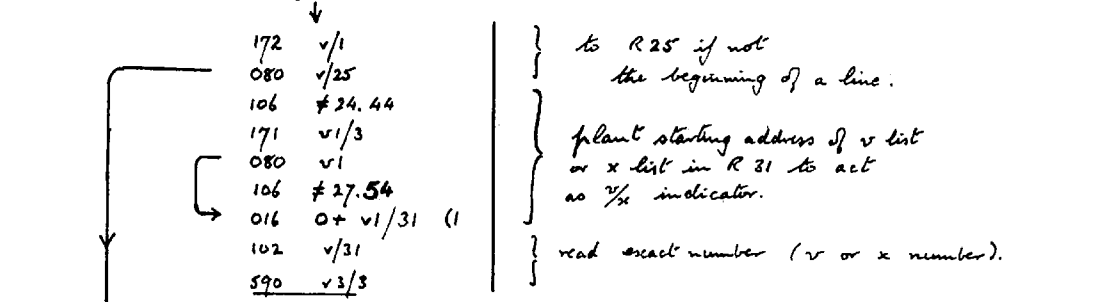
R 2

x Entry.



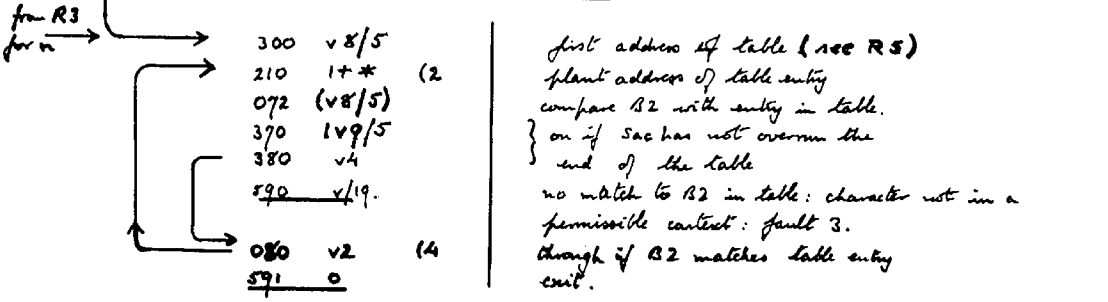
R 24

Separate v = , x =



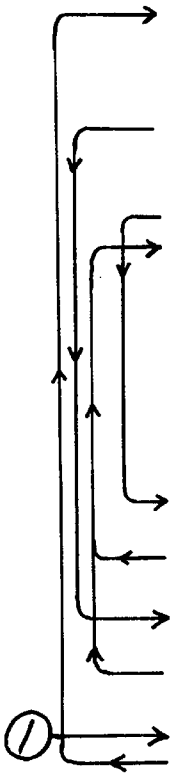
R 25

Test v, n, x or * O.K.



R 17.

/ Entry.



```
300  -1  (1        } set the / switch in R 23
210  0+v/23
590  v5/14         to clear the accumulator and return to read.

300  5   (2        } Sac = 5 - t
230  0+v/8
290  v/19          fault 3 unless t > 5, i.e. one, Π or L.
370  -1            } jump if Π or L, through if one.
280  v3

172  v/32          } in one, jump unless after fixed address.
080  v6
210  0+v/23 (4     set / switch : Sac = -1
300  256           } plant 'absolute address' indicator in R 3.
210  0+v2/3
010  43            clear l in H 43, i.e. refer to v 0.
200  0+v/8         plant t in reference
210  44+                 (8l+t with l = 0)
104  v2/3          } terminate function phrase as though a
592  0                   v reference had been read.

172  v/22 (6       } fault 3 except for x in one.
080  v/19
590  v4

172  v2/64 (3      } in Π or L : fault 3 unless M.
187  v/19                set Sac = -1.
590  v4.

172  v/23 (5       } jump if not after v, i.e. absolute address
080  v2                  in one or M.
101  v1            set return address and run on to store l.
```

R 26

Destandardise and store label number.

(Entered with l in the accumulator).

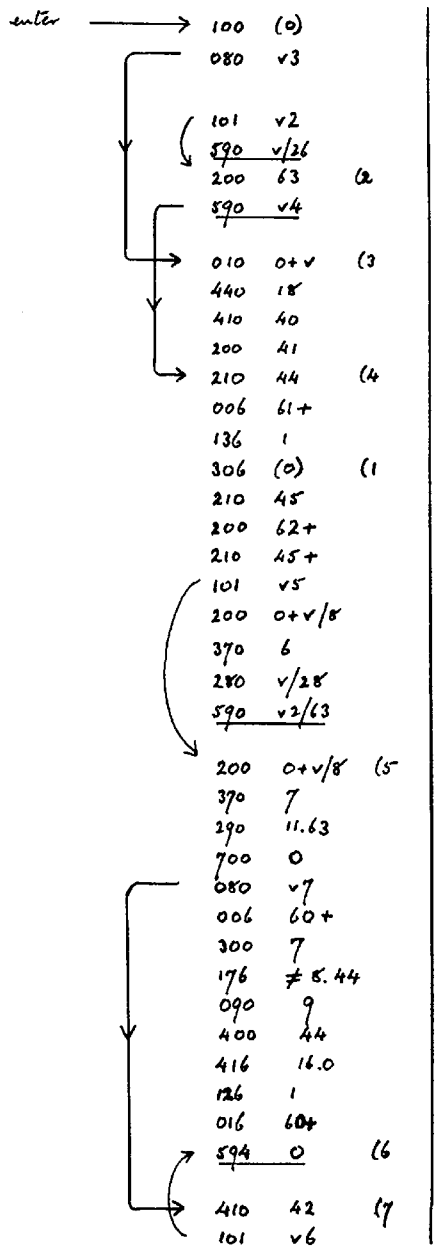
```
440  28
410  40            destandardise accumulator to 2^3.l
200  41            and plant in working space.
220  0+v/8         } plant 8l+t in reference.
210  44+
440  18            destandardise accumulator to l.
410  42            plant l in H 43.
591  0             exit.
```

July 1959.

R 23.

Terminate v or n Reference.

Entered from a "592 0" on the terminating character after the reference, with the return address in B4. Compiles a reference in L 44, and adds it to the list if the label referred to is not set.



/ switch: 0 if no /, −1 if / (set by R 17)
jump if / read (i.e. v in accumulator)
through if no / read (i.e. l in accumulator)
} destandardise and plant l.

v to Sac

clear / switch

} v from accumulator to Sac.

plant v in reference.

} 'h' of reference.

address part is v/n or abs. address indicator, planted by R 3.
plant indicators and h in reference.

} plant s in reference.

set return address for R 28

} St = t − 6

to locate label unless reference in a cue.
if cue, exit to complete cue reference.

} having located label, exit if M or L, both of which have left appropriate programme in pages 11 and 12.

} R 28 leaves the label in the accumulator (and L 42); test if label set and jump if so

label not set: pick up next position in reference list.

} fault 7 if too many references in list

} add reference to list

} step position indicator

exit.

(not necessary)
set return address and run on to fill in reference (R 27).

Reference: | v | 8l+t |
indicators + h | sector number.

July 1959.

R 27.

Fill in Reference.

Closed B1. Entered (from R 22, R 23, R 51) with the label in L 42, the reference in L 44, and the sector into which the reference is to be filled in page 1.

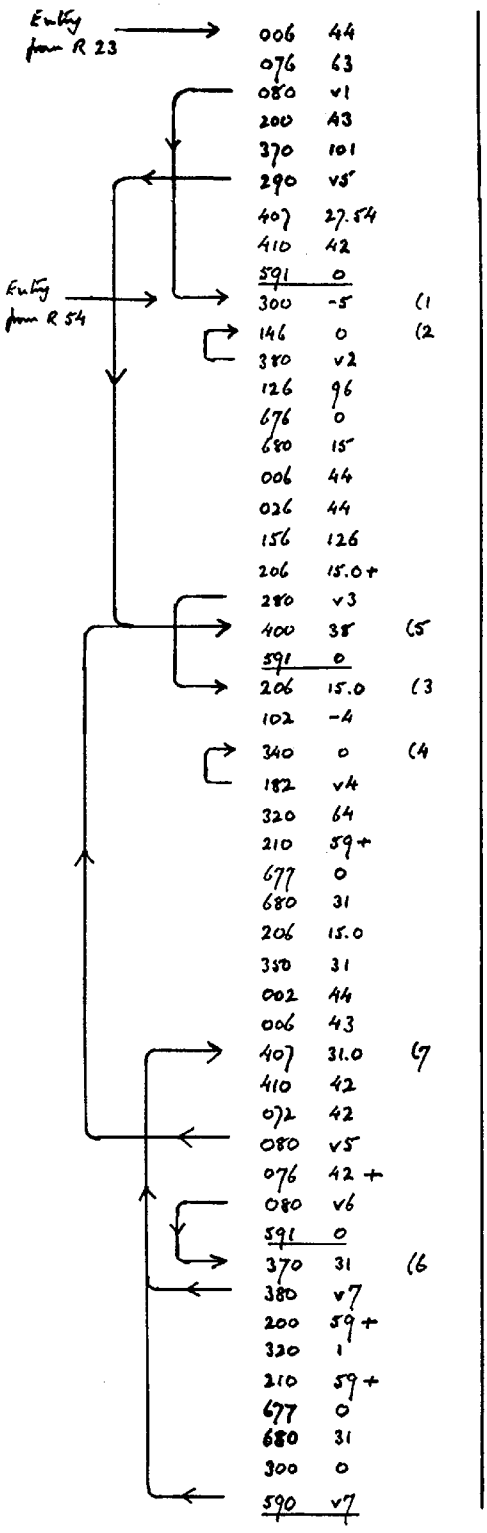
entry to fill in v, x, etc. →

	011	0+v6	plant return address.
	200	43	value of label
	210	23	
	007	43+	overflow bits
	350	15	
	210	23+	
	400	22	address including overflow to accumulator
	010	23+	
	090	v2	add ½ to accumulator if underflow bit present
	440	20	
→	101	v4 (2	set return address for R 8.
	200	44+	8l+t
	350	7	t
	370	3	to halve or double accumulator according to type, except for type 3.
	280	v3/8	
			type 3 only:
	006	43+	32i to 86
	156	224	
	176	32	
	300	4	to halve if i = 1 ('flad' set by floating point number label)
	080	v3	
	590	v3/8	
→	176	96 (3	to double if i = 3 ('flad' set by integer label)
	080	v4	
	380	v3/8	
entry from R8 →			
entry from R22 to fill in * →	006	45 (4	½n indicator and h (no alt. address indicator possible)
	090	3 *	jump if v
	440	v1/8	remove any fractional part of accumulator and negate
	520	2	
→	156	127	h
	206	1.0	pick up ten bits to which 'flad' is to be added.
	210	23	add fixed address as a floating point number
	440	22	
	440	18	
	410	40	plant filled in address.
	200	41	
	216	1.0	
	200	44+	t
	350	7	
	370	5	ignore overflow except for type 5
	280	v6	
	200	41+	overflow bit
	350	1	
	280	v5	to end if no adjustment of function required.
	590	v6	
→	206	63+ (5	adjust most significant function bit
	320	512	
	216	63+	
→	590	(0) (6	return.

July 1959

R 28
Locate Label.

Closed B1. Entered (from R 23, R 54) with the number of the required label in H 43 and the full reference in L 44. Puts the label into the accumulator and L 42. If the label is not set, the accumulator is left cleared to programme zero.



v from reference
} jump if label required is not of current routine.

l
if l ≥ 101, i.e. a quicky reference, treat as unset label (i.e. list reference).

} select label from list in the computing store.
(this space will be clear if the label is not set).
exit

} not the current routine: select the appropriate sector of the routine list (64 entries per sector) and bring to page 15.

} 2v to B6

} pick up from the routine list entry the number of the chapter containing the required routine.
if this is zero, the routine has not yet been read.
programme zero to the accumulator.
exit.
start of label list for this routine.

} sector number of start of label list for this routine.
store sector number.

} sector from label list to p. 31.

} position (within the page) of the first label of the required routine.
required routine number, v, to B2
required label number, l, to B6.

} label from list to accumulator and L 42.

} compare routine number in label with v. If these are different, the routine's label list has ended and the label not found.

} if the routine numbers check, compare label numbers. Exit if label located.

} cycle labels, testing for the end of the page.

} step sector number by 1, bring new sector of label list to page 31 and continue comparison from the start of page 31.

Reference: _ _v_ _ _ | _ _sl+t_ _
indicator + h | sector.

Entry in routine list:
address of first label from sector 64 | chapter number.

July 1959.

R 20.

Inspect h.

Closed B 1. Entered at v3 with h in B6, the routine plants h in H 61+ and then tests if h is 128. If not, it exits immediately. If so, page 1 is written to sector s (H 62+), s is advanced by 1, fp (H 62) is advanced by 1 and tested against 16 (to detect overflow of page 15), and H 61+ is cleared, i.e. h is set to zero ready to start the next page.

Entered at v4, the routine takes h from H 61+ and makes it even, i.e. sets h for the start of an instruction, one half register being skipped if necessary; entered at v, the routine takes h from H 61+ and makes it a multiple of four, i.e. sets h for the start of a floating point number, exec. routine, etc., inserting a dummy instruction if necessary. After either of these increases, h is tested as above for having reached the end of a page.

```
006  61+          (2     h to B6
126  1                   } next even h to B6
156  -2
306  0                   } test if h is a
350  2                     multiple of 4.
280  v2
590  v3
300  776          (2     } if not, plant '570' function
216  1.0                 } in the next register and increase
126  2                   } h to a multiple of 4.
016  61+          (3     store h
176  128                 } exit if h is not 128.
081  0
200  62+                 s to Sac
370  126                 } test s against 126 (in which case an interlude
380  v5                  } may be being read) and increase s by 1.
006  63+                 } if s was 126: test c against 127 to
176  127                   test if in interlude; if so, fault 8
080  v5                  } for interlude too long.
590  -2v4
677  -1           (5     } write up page 1 to old sector
690  1
210  62+                 plant new s.
677  0                   } new sector to page 1 (to preserve anything not
680  1                   } overwritten by new programme)
200  62                  } advance fp
320  1
210  62
006  58+                 } alter 'highest page so far' if necessary.
076  62
090  2*
210  58+
106  0                   } set h = 0
016  61+
370  16                  } exit if new fp is not 16
280  v3
300  8                   } fault 8 for chapter overrunning page 15
590  9
006  61+          (4     h to B6
126  1                   } next even h
156  -2                  to test against reaching the end of a page.
590  v3
```

July 1959

R 4.

Plant ten-bit word from Sac.

```
006  61+      h to Sac
216  1.0      plant Sac.
126  1        } shift h.
016  61+
591  0        exit.
```

R 36

Terminate *.

Entered from a '592 0' instruction after reading the next terminating character after *.

```
173  v/7      } fault 3 if decimal digit read after *.
080  v/19
101  v2/11    } arrange that R 27 finishes with a jump to a
011  0+v6/27  } '594 0' instruction, the standard return from a
              "terminate" routine.
101  -1v1/22  set B1 for eventual entry to R 22.
200  0+v/8    } if t = 7, i.e. the * is in a line directive, avoid
330  7        } subtracting Δ from h (since h will not have been
290  v1/33    } increased). Otherwise run on to v/33. (* in an
              Π directive will already have been caught in R 64).
```

↓

R 33.

Calculate * ; plant × 0.

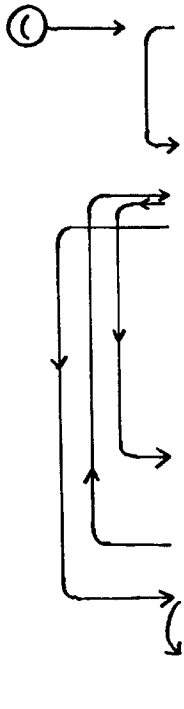
```
300  -1       } Sac = -1 for integer
000  0+v/18   }     = -2 for address
080  v1       } (0+v/18 is set to 0 for address, 1 for integer)
300  -2
220  61+  (1   h-Δ: * refer to half register of integer or
590  v1/11          first half register of instruction.
```

entry from R 11 for × 0. →

```
400  42   (2   from R 11 for × 0 (see R 50):
410  24.44      plant × 0
590  v/29       return to R 29 to restore chapter 1.
```

R 5.

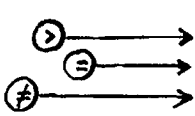
(Entry.



```
080  v1          jump if not at the beginning of a line.
670  40          at the beginning of a line, bring R 73
680  11          to page 11 and enter to ignore characters
590  v1/73       to ).
000  61    (1    set Bt = 0 if ( after quiche, ≠ 0 otherwise
300  v7          set Sac to address of first table entry used by R 5
080  v11   (4    } pass if i = 0 for quiche or if B2 checks
590  v5          } with a table entry.

                 Table used by R 5 and R 25.
= v2/64 (8,  = v1/49    after M ; after L
= v/32  (7,  = v/23     after absolute address or integer ; after v reference
= v/22   ,   = v/11     after x reference ; after label setting by (.
= v/36 (9,   = v/37      after *  ; after argument of f.p. number
= v2/38  ,   = v3/28 (3 after exponent of f.p. number ; after UT.

210  0+v2  (11   plant address of table entry
072  v7    (2    compare B2
370  v3          } cycle through table
380  v4

080  v/19        fault 3 if no entry in table checks with B2.
104  v6    (5    } terminate previous phrase.
592  0           }
102  v/11  (6    set B2 to terminate (.
590  v3/3        read label number.
```

R 13

>, =, ≠ entries.



```
320  2           } Sac = 2 for >
330  2           }       = 1 for =
320  2n          }       = 4 for ≠
006  0+v/8       previous type
172  v/1         reset Bt
210  0.+ v/8     plant type set by >, = or ≠
080  v1          jump except at the beginning of a line.
006  61+         } integer : inspect h.
101  v3          }
590  v3/20       }
300  1     (3    } set integer indicator in R 18.
210  0+v/18      }
300  3           } set item 3.
210  61          }
102  v/32        to read integer.
590  v3/3        }
172  v/32  (1    } jump if not in absolute address.
080  v2          }
176  6           } fault 3 if in error.
090  v/19        }
173  v/7         } fault 3 if d.d. read before >, =, ≠.
080  v/19        }
590  v/1         return to read with function type set.
172  v/31  (2    } fault 3 if not in equation.
080  v/19        }
430  v/40        } fault 13 if label number on the left hand
490  v1/40       } side of the equation is > 100.
440  v/40        }
440  18          } label number to M 47.
410  46          }
590  v3/3.       to read right hand side.
```

July 1959.

R 11

Terminate Label Set by (.

```
450  v/40
490  v1/40
440  v/40
440  18
410  40
200  61 +
006  61
236  32
101  v3
```

} fault 13 if label number > 100

} label number to H 41

Sac = L
B6 = i
L-Δ : correct L according to item.
set B1 for '591 0' below

from R 33, R 50, R 67 →

```
210  25 +  (1
400  24
200  62
210  27
440  26
171  -1v1/22
080  2*
590  v/8
```

} ½L + 64f to the accumulator.

} to R 8 to halve or double if in * (return to R 22).

```
440  18
410  42
200  63
210  42
206  35
220  42 +
220  43 +
330  384
210  43 +
591  0
```

} ½L + 64f destandardised to L 42.

} v to H 42

32 i
+ underflow
+ overflow and destandardiser
remove the destandardiser
plant in H 43 +
exit with label in L 42 for x0 or quicky.

```
006  41  (3
016  42 +
406  27.54
900  0
107  4   (4
280  9
400  42
416  27.54
594  0   (2
```

} plant l in label.

label

} fault 4 if label already set : used by R 50 for routine set twice and R 55 for chapter set twice.

from R 50, R 55. →

} plant label.

exit : used also by R 32.

R 30

, Entry.

```
172  v/37
080  v/6
```
} treat , as CR except in decimal input.
For decimal input, run on to read exponent.

↓

R 38.

Read decimal exponent.

↓

```
101  v1
590  v/84
410  40        (1
102  v2
590  v3/3
101  v3        (2
590  v/84
440  18        (3
410  42
200  0+v2/39
220  43
210  0+v2/39
400  40
```
} adjust argument for sign.
store argument.
} clear accumulator (in case no exponent read) and read exponent.
} adjust exponent for sign.
destandardise and plant exponent.
} add the exponent into the count of – the number of decimal digits after the decimal point.
replace the argument in the accumulator and run on to R 37.

↓

R 37.

Terminate number.

Entered either by run on from R 38 or by a "592 0" on reading the terminating character after the floating point number. Exit address in B4.

↓

```
101  v1
590  v/84
410  40        (1
200  0+v2/39
280  v2
590  v9
290  v5        (2
320  1
400  48
590  v4
500  48        (3
380  v3        (4
590  v8
300  1         (5
230  0+v2/39
400  16
590  v7
520  16        (6
380  v6        (7
820  40        (8
006  61+       (9
306  4
210  61+
146  0
146  0
416  1.0
594  0
```
} adjust decimal argument for sign if no exponent read.
If an exponent has been read, the sign switch has been cleared and R 84 has no effect.
store decimal argument.
decimal exponent (including adjustment for digits after the point)
} jump if exponent is zero.
jump if exponent is positive.
} $\left(\frac{1}{10}\right)^m$ to the accumulator
} $(10)^m$ to the accumulator
decimal argument × decimal exponent
} step h by 4.
} plant floating point number, using the old h.
exit.

July 1959

R 9.

Function and b - digits.

Entered at the first instruction from a 593 0 (see R 1) when a decimal digit at the beginning of a line is read, with the digit in Sac.

```
102   v/19          set B2 to give fault 3 if a terminating character is
                      read before the end of the function and b digits.
106   2             } set item 2 for instruction.
016   61            }
210   0+v1          }
327   0             } 10 x first decimal digit
327   (0)   (1     }      in Sac.
327   0             }
210   0+v2          plant.
101   v/1           set B1 for return to read.
103   v2            set B3 for return to R9 when next digit read.
590   v4/20         make b even + return to read next digit.

320   (0)   (2      second digit + 10 x first digit.
207   v/16          look up in table for binary form of function, and type.
210   0+v3          store
350   7             } isolate and plant type.
210   0+v/8         }
300   (0)   (3      restore function and type.
350   -8            function in top seven bits of Sac.
210   0+v4          plant.
103   v4            } set B3 for return to R9, and go to R1 to
590   v/1           }      read b-digit.

320   (0)   (4      add function bits to b-digit bits.
101   v3/3          set B1 for return to read exact number (i.e. address).
010   0+v/18  (5    set equation/address/integer switch to address.
102   v/32          set B2 for terminating fixed address.

590   v/4           to plant function and b, return to read address.
```

July 1959.

R 34.

Terminate and Fill in Preset Parameter on the Right Hand Side of an Equation.

Entered at the first instruction from a 592 0 when the first terminating character after the x is read, with the x number in the accumulator.

```
440  18      } destandardise and plant
410  40        x number.
006  41      } plant 'label' refering to r.h.s. parameter
406  24.44       in lines 42, 43.
410  42        32i  (= 128 for x parameter)
200  43+
280  3 *     } fault 6 if r.h.s. x not set.
300  6
590  9
006  0+v1/31   l.h.s. v/x switch
026  47         plus v/x number
406  0          select l.h.s. v/x
410  44         and plant in L 44
200  43       } r.h.s. x
210  23
007  43+      } overflow bits
350  15
210  23+
400  22         r.h.s. x with overflow to the accumulator.

090  2 *      } add ½ for underflow.
440  20
101  2 *      } to halve or double according to type
590  v/8        (set by ≠, = or > in the equation).
200  45       } value of left hand side v/x, not including
210  23         present parameter, to L 22 in floating
200  45+        point form
350  15
210  23+        add to r.h.s. x in accumulator
440  22         clear 23+
010  23+      } add previous underflow
200  45+
290  2 *
440  20
440  18       } destandardise and plant new value of l.h.s. v or x.
410  40
200  41       } central ten bits
210  45
200  45+      } present 32i in l.h.s. v or x.
350  480
220  40+        add underflow bit
220  41+        add overflow and destandardising bits
330  384        remove destandardising bits.
210  45+        plant.
400  44         new 'label' for l.h.s. v or x.
006  0+v1/31  } position of l.h.s. v or x.
026  47
416  0          plant new 'label'.
594  0          exit.
```

from R 80 (L, P, S on r.h.s. of equation)

July 1959

R 39.

Number Input.

Entered at the first instruction from R 12 if + or − starts a line.

```
010  0+v2          } clear exponent count'
010  0+v3          } and decimal point switch.
106  1             } set item 1 for number.
016  61            }
101  v/1           set B1 for return to read.
400  0      (1     clear accumulator.
103  v5            set B3 for first digit of number.
102  v/37          set B2 for "terminate number" routine.
590  v/20          to set even register ; return to read decimal digits.
103  v2     (5     set B3 for subsequent digits of number.
106  (0)    (2     exponent count : minus number of digits after decimal point
126  (0)    (3.    decimal point switch : 0 if no point, −1 if point.
016  0+v2          reset exponent count : 0 before point, stepped by −1 after point.
520  16            } accumulator × 10 + new decimal digit.
440  14            }
590  v/1           return to read.
300  −1     (4     } decimal point in number entry; set
210  0+v3          }   decimal point switch.
590  v/1.          return to read.
```

from R 14 (point in number) →

R 31.

Terminate Absolute Part of Right Hand Side of Equation.

Entered at the first instruction from a 592 0 when first terminating character on the right hand side of an equation is read.

```
300  −1            } plant' equation indicator in R 18.
210  0+v/18        }
590  v/32          enter R 32 to adjust address for sign, etc.

106  (0)    (1     % switch : ≠ 24.44 if x, ≠ 27.54 if v. (set by R 24).
026  47            plus number of x or v on right hand side.
300  4
176  379           } jump if x.
090  1v3           }
406  0             } fault 4 if v label set twice.
410  42            }
000  43+           }
080  9             }
200  63     (3     } v                                  } compile label
210  42            }                                    } in L 42.
200  47            } v or x number                      }
210  42 +          }                                    }
200  41            } medium address                     }
210  43            }   on v.h.s.                        }
200  41 +          } overflow                           }
350  15            }                                    }
320  128           + 32 i with i = 4, (equation set)    }
220  40+           + underflow                          }
210  43+           }                                    }
400  42            } plant' label.
416  0             }
590  0             out'.
```

from R 32 →

July 1959

R 16.

Function Table.

One entry per function, listed in order of decimal function code. Each entry is 10 binary digits, 7 for the binary form of the function and three for the function type. The entries for functions with different binary equivalents for reference to the first and second quarters of the store are the forms refering to the first quarter.

S24

6 = 333, = 277
7 = 325, = 365
(= 357, = 349
: = 341, = 373
= 57, = 129

= 75, = 512
= 67, = 107
= 99, = 91
= 83, = 115
= 121, = 512

= 461, = 405
= 453, = 493
= 485, = 477
= 469, = 501
= 185, = 257

= 203, = 512
= 195, = 235
= 227, = 219
= 211, = 243
= 249, = 512

= 268, = 284
= 292, = 300
= 308, = 316
= 820, = 828
= 393, = 137

= 420, = 428
= 436, = 444
= 180, = 52
= 512, = 776
= 384, = 9

= 21, = 149
= 681, = 170
= 412, = 512,
= 512, = 33
= 161, = 41

R 16

Function Table (continued).

= 584, = 152
= 576, = 616
= 608, = 600
= 592, = 624
= 800, = 512

= 697, = 673
= 553, = 544
= 512, = 512
= 908, = 897
= 649, = 641

= 712, = 760
= 704, = 744
= 736, = 728
= 720, = 752
= 512, = 0

S 24

192	128
448	81
400	87
264	21
286	0
12	0
333	277
325	365
357	349
341	373
57	129
75	27
67	107
99	91
83	115
121	512

461	405
453	493
485	477
469	501
185	257
203	539
195	235
227	219
211	243
249	512
268	284
292	300
308	316
820	828
393	137
420	428

436	444
180	52
512	776
384	9
21	149
681	170
412	512
512	33
161	41
584	152
576	616
608	600
592	624
800	512
689	673
553	544

512	512
908	897
649	641
712	760
704	744
736	728
720	752
512	

R 16

Function Table (Continued).

= 584, = 152
= 576, = 616
= 608, = 600
= 592, = 624
= 800, = 512

= 697, = 673
= 553, = 544
= 512, = 512
= 908, = 897
= 649, = 641

= 712, = 760
= 704, = 744
= 736, = 728
= 720, = 752
= 512, = 0

July 1959.

LINE 14.0

The remaining routines of Chapter 1 may be required during the action taken for directives, etc., at a time when pages 11-13 of Chapter 1 have been overwritten. R 32 therefore begins at line 14.0.

R 32.

Terminate Absolute Address or Integer.

Entered at the first instruction from a 892 0 when the first terminating character is read in the address of an instruction or cue, or in an integer; also from R 31 during termination of the absolute part of the right hand side of an equation. The absolute part just read is in the accumulator.

101	v1		return address from R 84
440	50		add any stored part of absolute address (e.g. page number)
590	v/84		to adjust absolute address for sign.
101	v2	(1	} to halve or double according to type.
590	v/8		
440	18	(2	} destandardise and plant absolute address.
410	40		
400	0		} clear accumulator and
410	80		address storage space.
000	0+v/18		equation/address/integer indicator.
090	2 *		} return to R 31 if r.h.s. of equation
590	v1/31		(indicator = -1)
000	40 +		}
300	1		} fault 1 for address underflow.
080	9		
200	41 +		} overflow, without destandardiser bits,
330	384		to Sac
101	v3		} if overflow, enter R 18 to test if
280	v/18		allowed.
200	41	(3	absolute address
101	v2/11		address of a "594 0" for return from R4;
590	v/4		enter R4 to plant absolute address or integer.

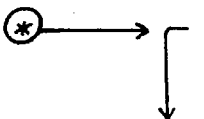
R 84

Adjust the accumulator for sign.

100	(0)	sign switch
091	0	out if positive
320	2	negate accumulator
010	0+v	clear sign switch
591	0	out.

R 35.

* Entry.



```
080  v1
000  58
090  v/19
010  58
590  v/1
104  v3      (1
101  lv1/3
590  v/25
102  v/36    (3
590  v3/3
```

jump if not at the beginning of a line.
* switch : −1 when * printing not set', 0 when set'.
fault 3 if * printing set twice.
set * printing switch
return to read.
set B4 for return from terminating previous phrase.
set B1 to address of standard "592 0" to terminate previous phrase.
to test if * O.K., return to a "592 0" instruction.
from terminating previous phrase; set B2 to terminate *
return to read exact number.

R 22

Terminate x in Address.

Entered at the first instruction by a "592 0" when the first terminating character after the x is read. The x-number is in the accumulator. Also used by *.

```
440  18
410  40
006  41
406  24.44
410  42
200  43 +
101  v2/11
```

} destandardise and plant x number.

x number
} selected parameter label to L 42 (as required by R 27)

32i to Sac.
set B1 to address of a "594 0" instruction, to make R 27 effectively closed B4

```
280  v1
300  6
590  9
```

} jump if parameter set'
} fault 6 if parameter not set'.

entry from R 33 for *

```
100  0
200  0+v/8   (1
370  7
```

set Bt = 0
type
} jump if not M or L.

```
280  v2
080  11.63
440  18
410  42
200  42 +
220  43 +
210  43 +
590  11.63
```

to 11.63 if x in N or L, through if *.
destandardise current address (set in accumulator by R 33)
plant destandardised address in L 42 as a 'label'.
} adjust for underflows.

to 11.63 for * in M or L.

```
210  44+     (2
200  61 +
330  1
210  45
080  v/27
590  v4/27
```

plant t } t and h of reference in L 44 as required by R 27.
h }

to fill in parameter.
to fill in *

CHAPTER 2.

(Sectors 26 - 28).

C 2 P 11

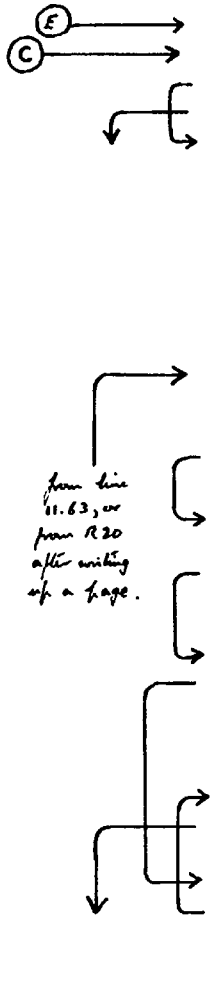
R 53

(Sector 26)

C, E Entries.

Occupies page 11. Brought to page 11 and entered from R 87 in page 15. Terminates the last chapter, writing the final pages to the drum, and enters the routine to fill in exits for C. Reads chapter number for C entry.

The routine is also used by M, which like C terminates the previous chapter. In this case, R 53 is brought to page 11 by R 64 (also in page 11) and entered at the second instruction with B2 set to v/65.




```
102  v/59      set B2 for E
006  63+       previous chapter number.
080  2 *       } jump if zero, i.e. no previous chapter.
590  v8        }
176  127       } fault 3 if in interlude.
090  v/19      }
670  28        } half of R 51 to page 13.
680  13        }
101  26        set B1 for return to this sector.
010  0+v17/51  clear indicator for filling in of quickies.
590  v4/51     enter R 51. (Return to line 11.63).

670  113  (1   from line 11.63:
006  63+       bring to page 15 the sector of the chapter list
176  64          appropriate to the old chapter, and reset
090  2 *         B6 to the old chapter number.
670  112
680  15
200  57   (11  predicted last page of old chapter
280  v3        } if no predicted last page set, set it equal
300  58+       }   to the highest page read.
210  57
270  58+  (3   } fault 9 if highest page read is
290  v4        }   greater than the predicted last page.
300  9         }
590  9         }

101  v1   (5   } set B1 for return to write up more pages
590  2v8       }   of the old chapter

270  62   (4   } compare current page with predicted last page;
280  v5        }   through if only the last page of the chapter remains to
                   be written up.
```

July 1959.

R 53 (Continued).

(occupies page 11).

200	62+	s to 5ac
026	63+	2c to 86
156	126	modulo 128
226	15.0+	add first page (f_1) from chapter list to s.
230	62	$s + f_1 - f$ = first sector of chapter.
216	15.0	plant in chapter list
206	15.0+	
105	-4	} f_1 shifted up 5 places
327	0 (7	
185	v7	
220	62	add f_2
216	15.0+	plant in chapter list
690	15	write up chapter list.
010	62 (8	clear f
101	v9	return address after writing up last page.
106	128	set 86 = 128 to make R 20 write up page 1.
590	v3/20	write up page, return to v1 or v9.
		(leaves $f=1$, $h=0$ after writing up last page).
010	57 (9	clear predicted last page.
017	58+	set highest page read to 1.
670	30	} bring down R 54 to page 12.
680	12	This also brings down R 65, required by M.
172	v/1	} to bring R 59 to page 11 if E or M.
080	11.61	
101	v10	} fill in rules if C.
590	v/54	
670	33 (10	} bring to page 12: R 55 ("Terminate C") and
680	12	R 56, R 58 (P and S entries)
102	v/55	set 82 to terminate C.
590	v3/3	read new chapter number.
LINE	11.61	
670	29	E or M:
680	11	} read R 59 to page 11, running on to a "592 0" in
		line 11.63. Enter R 59 for E, or R 65 for M.
590	v1.	Return from R 51.

Chapter list entry (20 bits):

During reading of the chapter: 0 , f_1

After terminating the chapter: first sector , $32 f_1 + f_2$

R 51

(Sectors 27, 28).

References and Labels.

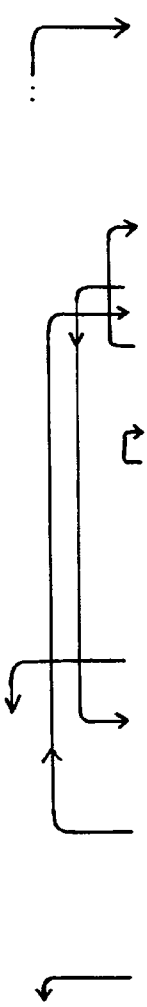
Occupies pages 12 and 13. Entered at v4 (line 13.0) at the end of each routine or chapter. Return by bringing the sector set in B1 to page 11 and entering it at 11.63.

R 51 fills in all outstanding references to labels of the last routine (so long as those labels are set), controls the insertion of quickies for references to labels 101 - 119, and packs the remaining references into page 16 onwards to leave no gaps. R 51 also packs the label list of the last routine (in the computing store from 27.54 onwards) and writes up the list to the next position in the label list on the drum. It leaves the computing store clear from 27.54 to the end of page 30 ready for the next routine's label list.

```
011   0+v23   (1     plant return sector
017   0+v3            plant last routine number.
360   -31             } clear page 31 for 'labels' to inserted quickies.
400   38              }
417   31.62           }
380   -1*             }
105   0               set B5 to work through reference list.
200   62+     (24     } write up page 1 (re-enter here with B5 equal to
677   0               }   position of quicky reference after filling in quicky).
690   1               }
590   v9.             to scan references.

101   v21     (19     from planting '591' function; set B1
005   47              quicky number to B5
300   0               }
175   9               } to plant '3' for quicky 9
080   v14             }      '0' for other quickies.
300   3               }
590   v14             }

400   42      (21     } set 'label' for quicky in page 31.
415   30.62           }
002   13.0            } restore B2, B5
005   13.0+           }
590   v24             re-enter reference loop, writing up current page 1.

410   42      (5      label set: plant label in L 42
200   45+             } sector to page 1.
677   0               }
680   1               }
101   v6              } to fill in reference
590   v27             }

690   1       (6      write up sector with reference filled
590   v8              to test for end of reference list.
```

R 51 (continued)

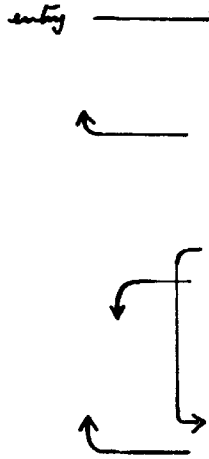
(Pack and clear labels).



```
176  32  (22      } fault 12 if too many labels.
090  9
016  59+          store sector indicator in H59+
676  64           } next sector of label list to page 31
680  31           }
006  59           } next position in label list (modulo 32)
156  31           }
405  27.54 (11    next label
707  0            } jump if set, ignore if not
080  v15          }
175  100   (12    } test for 100 labels
185  v11          }
200  59+          } next position in label list (relative to the
105  -4           }   start of sector 64) in H 59.
327  0     (13    }
185  v13          }
326  0            }
210  59           }
690  31           write up last sector of label list used.
200  62+          } restore current sector to page 1.
677  0            }
680  1            }
590  v23          to exit.

416  31.0  (15    fault set label in page 31.
400  38           } clear computing store label.
415  27.54        }
176  31           } test for page 31 full.
186  v12          }
690  31           write up page 31
010  59           clear position in page
006  59+          } 36 = next sector of list.
126  1            }
590  v16          back.
```

R 51 (Continued)

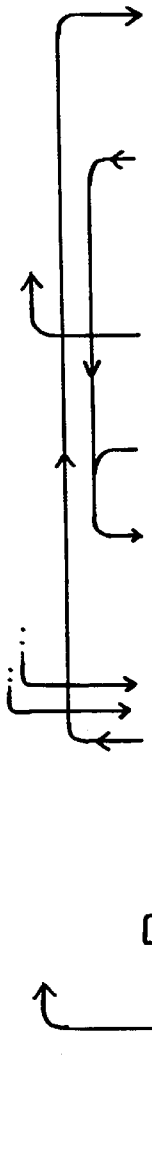
LINE 13.0



```
entry ──────►  670  27          } rest of R 51 to page 12.
               680  12
               400  27.54       first label of last routine.
               707  0           } to v1 if set, out if not set
               080  v1          }   (e.g. entry on first R directive of a chapter).
               590  11.63

               175  (300)  (17  } test position in reference list to see if quicky should
               090  2*         } be filled in :
               590  v7              0+v17 :  0   for C entry
                                         :  200 for R entry
                                         :  first position in reference list relevant
                                            to the interlude for J entry.

               700  0           } to fill in reference if quicky label is set (i.e. quicky
               186  v5          }   already incorporated). Add 1 to B6 to give quicky number.
               670  41          } quicky not incorporated: first part of quicky
               680  11          }   routine (R 67) to page 11.
               016  47          plant quicky number.
               101  v18/67      set B1 for entry to R 67.
               011  0+v16/67     set 0+v16/67 non-zero for return to R 51.
               012  13.0        } preserve B2, B5
               015  13.0+       }
               200  62+         } restore current sector
               677  0           }   to page 1.
               680  1
               590  v/20        to make register even: return from R 20 to R 67.

from R67 ──►   670  27  (20     } from R 67: restore the rest of R 51.
               680  12
               300  9           } to plant '591' function
               101  v19         }   after the quicky.
               590  v/4
```

July 1959.

R 51 (Continued).



```
405  16.0 (2      reference from list
410  44           plant in L 44
200  AA+          Bl+t
350  7            } ignore reference for t = 6, i.e. end.
370  6            }
290  v7           Bl+t
006  AA+          } l - 101 to B6
146  0            }
146  0            }
146  101          }
406  31.0         pick up label into the accumulator
090  v17          jump if l ≥ 101 with quantity number -1 in B6

200  AA           v to Sac
370  (0)  (3      } jump if reference not to last routine
280  v7           }
700  0            } jump if label not to fill in reference.
080  v5           }
400  44   (7      label not set: reference to accumulator
300  (0)          next position in new reference list (zero first time)
417  16.0         replace reference in new list
320  1            } step position in new reference list.
210  1+v7         }
125  1    (8      step position in old reference list
075  60+  (9      } test for end of references.
080  v2           }
200  1+v7         } plant new 'next' position in reference list.
210  60+          }
105  0            B5 = 0 for working through labels.
006  59           'next' position in 'label list'
300  -4           } B6 = number of sector containing the next
146  0    (10     }      position in the label list.
380  v10          }
300  (12) (16     Sac = 12 in preparation for "fault 12"
                    or zero for J entry
280  v22          to v22 except for J entry

670  (0)  (23     } exit to 11.63 of sector planted
680  11           }      from B1.
590  11.63        }
```

July 1959

CHAPTER 3

(Sectors 29 - 31)

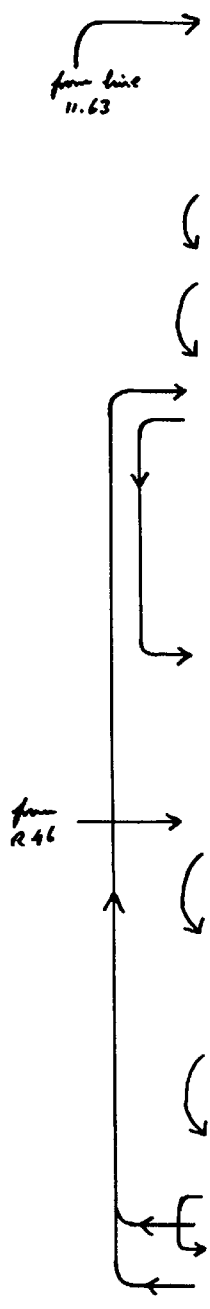
C 3 P 11

R 59

(Sector 29).

Terminate E

Brought into page 11 by R53; entered via a '592 0' in 11.63 with B2 set to v/59.




```
106  127
016  61+
200  57+
210  62+
677  0
680  1
101  v2
590  v/63
690  1        (2
101  v3
590  v/54

105  0        (3
075  60+      (6
080  v4
200  57+
677  0
680  1
670  3
680  0
590  59

670  35       (4
680  12
405  16.0
410  44
590  v/46

200  45+      (5
101  2*
590  v/47

620  28
200  45
350  126
340  0
101  2*
590  v/47

200  45
857  1
185  2*
590  v6
620  26
590  v6
```

set h and s to indicate H 63+ of the 'first sector', which is where the entry address is stored

'first sector' is page 1 for insertion of absolute part of the entire one.

read entire one: R 63 has been left in page 12 by R53, having been brought down with the start of R 54

write up first sector.

fill in ones, including the entire one, to which there is now a reference in the reference list.

set B5 for counting through references not set

cycle through unfilled references

first sector is page 1

enter sector 3 at line 59

R 47 ("Punch Sac") and R 46 ("Punch NOT SET") is page 12.

reference is L 44 to punch "NOT SET"

from R 46: is punch 5.

punch "."

punch line

punch "+" if h in reference is odd; step B5 and return to test it against 60+.

July 1959.

R 70

? Entry.

(Sector 29)

Brought to page 11 and entered at the first instruction from R 81.

```
(?) →  102  2*      } to read ? number.
       590  v3/3

       440  18      } destandardise
       410  40        and plant ? number.
       200  41      ? number to Sac.
       370  11      } jump if ≥ 11
       290  3*
       210  56      plant ? number
       590  v/29    to restore chapter 1 and return to read.

       300  14      } fault 14 for ? number bigger than 10.
       590  9
```

n printing quicky.

Inserted in the programme by R 67 (entered from R 71) an quicky 0 to give n printings. The quicky starts in line 53, the indication in line 51 (see notes to R 67).

```
= 0, = 7
= 76*, > 61         } indicator.

670  479            } store page 0
690  0
670  43             } R 72 to page 0
680  0
210  0+             plus twice relative address (i.e. ten bit address).
210  8+             plus twice relative address (i.e. ten bit address)
300  8              plus relative address (to give return address)
590  v1/72          enter R 72
300  0              restore Sac.
```

LINE 11.63

```
592  0
```

R 63.

(Sector 30)

Read Address in Cue.

Closed B1. Brought to page 12 and entered by R 60 (in page 11) for an Across or Down cue; for an Enter cue, brought to page 12 by R 53 but entered from R 59.

```
enter ——→   011  0+v1        } plant return address
            300  6           } set type 6 for cue.
            210  0+v/8       }

            102  v/32        set B2 for terminating absolute part of address.
            590  v3/3        read address.
                             The fixed part of the address is read and terminated as for
                             an instruction. During termination of the symbolic part in
                             R 23, type 6 is detected and R 63 re-entered at v 2.
from R 23 ——→ 006  60 +  (2  next position in reference list.
            400  44          reference, compiled by R 23
            416  16.0        list reference to cue.
            126  1           } step position in reference list indicator
            016  60 +        }
            590  (0)    (1   return.
```

R 65

(Sector 30)

(i) Restore R 64 to Page 11; (ii) Terminate M.

(i) Entered at the first instruction from the "592 0" at the end of R 59. The M routine, R 64, is obeyed in page 11, but it first enters the C routine, R 53, with B2 = v/65, so that the preceding chapter is terminated. R 53 brings R 65 into page 12 in the process of bringing down the start of R 54. R 53 then brings down R 59 and runs on to the "592 0" instruction in line 11.63, thereby in this case entering R 65, which restores R 64 to page 11 and re-enters it.

(ii) Entered again at v1 or 1v4 from R 64 to complete the M action.

```
from R 59 ——→  670  39        } restore R 64 to page 11
               680  11        }
               590  v1/64        re-enter R 64.

from R 64
(via R 54) ——→ 210  0+v4   (1   re-entered with Sac = first sector of chapter − first page of chapter
               200  51          new address for start of correction
               106  -5        }
               340  0         } page for start of correction
               186  -1 *       }
               210  62          plant new p.
               320  (0)    (4  first sector − first page + new p = new s.
from R 64 ——→  210  62+         plant new s. (Entered here for absolute M with Sac = new s)
               677  0           } sector to be corrected
               680  1           }     to page 1.
               590  v/29        restore C1 and return to read
```

R 68

(Sector 30)

Terminate Cue.

Brought down to page 12 by R 61. In the case of an UP cue, the return address for R 61, set by R 62, is v/68, so that R 68 is entered immediately the cue has been compiled. In the case of an ACROSS or DOWN cue, R 68 is entered at v1 from R 60, with Sac set = 5.

entry from R 61 for U.

```
102   v3
s90   v3/3

300   4      (3
210   61     (1
300   v2
210   0+v/6
174   v1/6
084   0

300   v1/6   (2
210   0+v/6
s90   v/29
```

entry from R 60 for A, D.

} return to read — CR or (expected.

item 4 for UP
plant item number, 4 for Up, 5 for Down or Across.
} change R 6 so that CR, after terminating previous phrase, will return to R 68.
} if the terminating character at the end of the cue was not CR, return to complete the required action. CR will eventually produce a return to v2.
} restore R 6 to its original form

restore pages 11 - 14 of Chapter 1 and return to read new item.

R 54

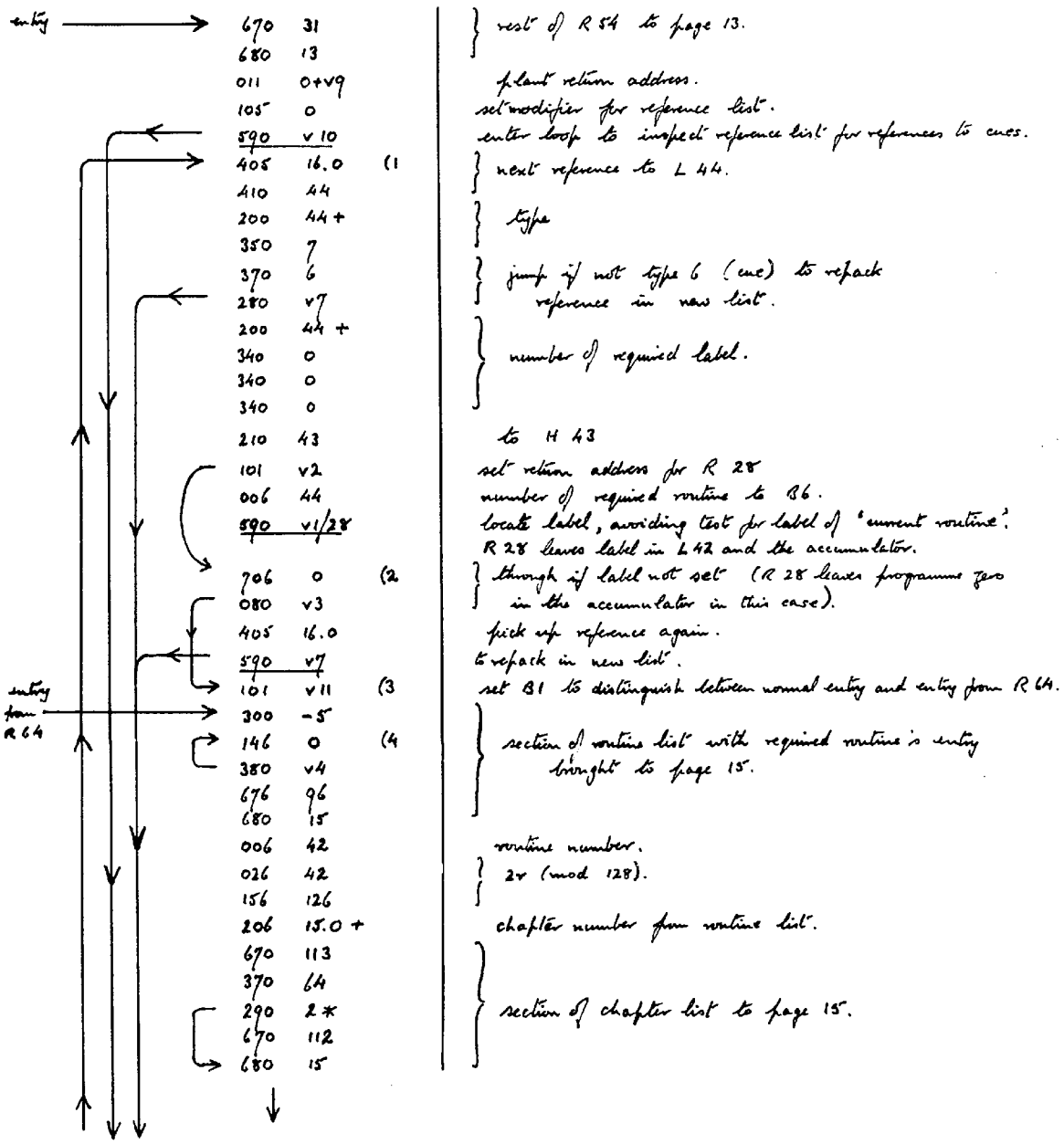
(Section 30, 31)

Fill in Cues.

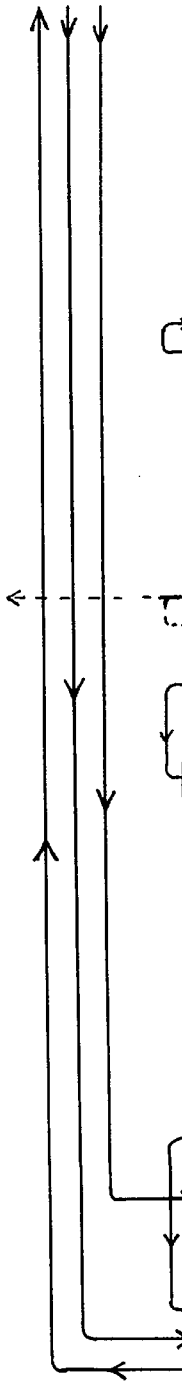
Closed B1. Obeyed in pages 12 and 13. The first part is brought to page 12 by R 53, and the routine is entered from R 53 for C, R 59 for E and R. 64 for M, so that all possible cues are filled in at the end of each chapter.

The routine plants the four integers $s_1 - p_1$, p_1, p_2 and e in an across or down cue. (The down/across indicator and the absolute part of e have already been planted, on the reading of the cue.)

The part of the routine which calculates $s_1 - p_1$, and brings to page 1 the sector required by the reference to the cue, is used again by the M routine in setting up initial conditions according to its own address, by entry at 1 v 3.



R 54 (Continued).



327	0		} 2c (mod 128)
350	126		}
210	40		store
207	15.0+		} $32f_1 + f_2$ from chaffer list
210	40 +		stored in H 40+
350	31		} f_2 stored in H 41
210	41		}
200	40+		} form f_1 and store
106	-4		in H 40+.
340	0	(6	}
186	v6		}
210	40+		}
200	45+		sector number from reference.
677	0		} sector to page 1.
680	1		}
200	40		2c to sac.
207	15.0		s_1 of chaffer from chaffer list.
230	40 +		$s_1 - f_1$
591	0		out if second M entry to v1/15; straight on for normal entry.
006	45	(11	indicators and h from reference.
136	256		}
090	3 *		} clear address in label
006	45		} if "absolute address in cue" indicator
590	2 *		} present. Leave B6 = h.
010	43		}
216	62+		plant $s_1 - f_1$ in cue.
206	63		} down/across indicator
350	512		} from cue.
220	40 +		add f_1
216	63		and replant in cue.
200	41		} plant f_2 in cue.
216	63+		}
200	43		value of label, i.e. symbolic part of e.
226	64		} add in to absolute part of e.
216	64		}
690	1		write up sector with cue completed.
590	v8		avoid repacking reference in list.
300	(0)	(7	label not set: next free position in new reference list to Sac.
417	16.0		list unfilled reference.
320	1		} step position in new reference list.
210	0+v7		}
125	1	(8	increase modifier for old list.
075	60 +	(10	compare modifier with end of list indicator
080	v1		back if more references.
200	0+v7		} plant indicator for end of new list.
210	60+		}
200	62+		} restore current sector to page 1.
677	0		}
680	1		}
590	(0)	(9	exit.

CHAPTER 4.

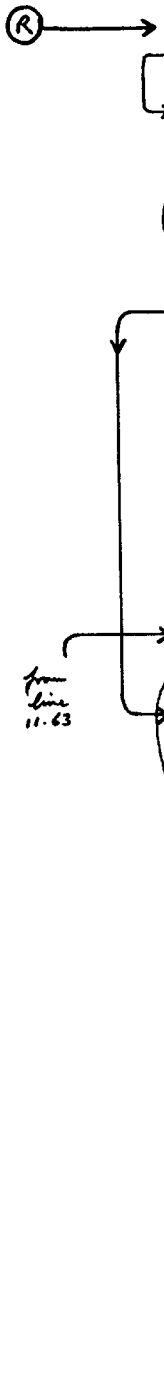
(Section 32, 33).

C 4 P 11

R 50

(Sector 32)

R Entry



```
R →  000  63+          } fault 3 if chapter not set.
     080  2*           } make register even.
     590  v/19         } read routine number v.
     101  2 *
     590  v/20
     102  v2
     590  v3/3

     450  30   (2      v - 1000
     101  v2/33        set B1 for return from R11 to set x0
     490  v6           jump if v ≥ 1000   (x - Routine)
     440  30           restore v
     440  18
     410  40           } new v to H 63
     200  41
     210  63
     670  28           } bring down part of R51
     680  13
     101  32           set B1 to return to this sector
     590  v4/51        enter R51 to fill in references

     010  41   (3      from R51 (via line 11.63). Set H 41 for v0
     104  v4           set B4 for return from R11.
     101  v3/11        set B1
     106  4    (6      set B6 for item 4  (v0 or x0 equation set)
     200  61+          L to Sac
     590  v1/11        enter R11 to set v0 or x0.

     006  63   (4      return from R11 for v-routine, v to B6
     300  -5
     146  0    (5      } sector of routine list to page 15.
     380  v5
     676  96
     680  15
     006  63           } plant address of first label
     026  63             in routine list.
     156  126
     200  59
     216  15.0
     206  15.0+        } to give fault 4 if the routine is already set.
     280  v4/11
     200  63+          } plant chapter number in routine list.
     216  15.0+
     690  15           write up new entry
     000  58           * printing switch
     080  v/59         to restore chapter 1 if no printing required.
     670  37           } R66 to page 13.
     680  13
     590  2.0 v/66     enter R66 at the first instruction.
```

from line 11.63

July 1959.

R 667
I Entry. (Sector 32).

(I) →	200	62+	s to Sac
	677	0	} write up current sector
	690	1	to the drum.
	670	127	} store page 0 on sector 127
	690	0	
	300	127	} set chapter number to 127
	210	63+	
	010	61+	clear h
	300	1	} set f and "highest page so far"
	210	62	to 1.
	210	58+	
	300	114	} set s to 114
	210	62+	
	677	0	} bring new current sector (i.e. 114)
	680	1	to page 1.
	590	v/29	to restore chapter 1 and return to read.

LINE 11.63

←	590	v3/50	return from R 51

R 55

(Sector 33)

Terminate C.

Brought down to page 12 by R 53. Entered at the first instruction from a "592 0" when the terminating character after the chapter number is read.

```
entry ──►  440  18          } destandardise and plant
           410  40            new chapter number, c.
           006  41          c to B6
           090  3 *       }
           300  13   (4     } fault 13 if c > 101.
           590  9         }
           176  101       }
           090  -3 *      }
           016  63+         plant c.
re-entry
after ──►  670  113  (1     } section of chapter list for new
1st setting.  176  64       } entry to page 15.
           090  2 *        }
           670  112        }
           680  15         }
           006  63+        } 2c (mod 128)
           026  63+        }
           156  126
           206  15.0         first half of chapter list entry.
           280  v4/11        fault 4 if chapter already set.
           200  62           plant first page of chapter
           216  15.0 +       in chapter list.
           690  15           write up chapter list with new entry.
re-entry after
reading last ──►  174  v1/6  (3   } if the terminating character was not CR, return to
page or sector  084  0           } complete action (must be page or sector setting)
           006  58           } back to restore Chapter 1 if no * printing required.
           080  v/29         }
           670  37           } R 66 to page 13
           680  13           }
           300  3            } enter R 66 at the third instruction
           590  2.2 v/66     } with Sac = 3 for C punching
```

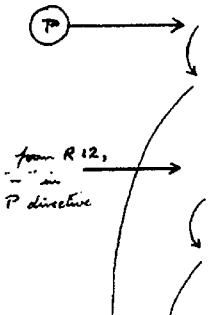
Chapter list entry while chapter is being read: 0 ⋮ first page.

R 56

(Sector 33)

P after chapter setting Entry.

Brought down with R 55 to page 12 by R 53, and entered at the first instruction when P is read.



```
P ──►  104  v1          } terminate previous phrase
       592  0           }   (chapter or sector setting).
       102  v/57  (1    } read first page f_f
       590  v3/3
from R 12, "–" in P directive ──►
       010  0+v/84 (2    entry on reading "–" after first page number: clear sign switch.
       104  v3          } terminate f_f
       592  0           }
       102  v1/57 (3    } read last page f_e.
       590  v3/3
```

R 57.

Terminate P after chapter setting.

Brought into page 12 with R 55, R 56.

```
       440  18   (1     } page number just read to Sac.
       410  40          }
       200  41
       100  (-1) (2     f_f/f_e switch: address part = -1 for f_f and = 0 for f_e.
       080  3 *         jump if f_f.
       210  57          plant f_e
       590  v3/55       terminate chapter action.
       010  0+v2        set switch in preparation for f_e.
       210  62          } current page and highest page so
       210  58+         }   far set = f_f.
       006  63+         } to alter chapter list entry and
       590  v1/55       }   terminate chapter action.
```

R 58.

S after chapter setting Entry.

Brought down with R 55 to page 12 by R 53, and entered at the first instruction when S is read.

```
S ──►  104  v1          } terminate previous phrase
       592  0           }   (chapter or page setting)
       102  v2   (1     } read sector number s_1.
       590  v3/3
       440  18   (2     }
       410  40          }   s_1 to Sac.
       200  41          }
       220  57 +        s = s_1 + first sector number.
       210  62+         plant as current sector.
       677  0           } current sector to page 1.
       680  1           }
       590  v3/55       terminate chapter action.
```

CHAPTER 5

(Sectors 34, 35)

C5 P 11
R 60

(Sector 34)

A and D Entries.

Brought down to page 11 and entered from R 81.

A →	010	0+v3	set switch in 0+v3 to indicate across.
D →	101	v2	(1 } make register even.
	590	v/20	}
	101	2*	(2 } build and plant first half of cue
	590	v/61	} (R 61 also brings R 63 and R 68 to page 12).
	101	2*	} inspect h in case the cue overlaps the end of a page.
	590	v3/20	} (R 61 leaves h in B6).
	300	(512)	(3 } plant D/A indicator in the f_4 half register of the cue.
	216	1.0+	} (sign bit for down, no sign bit for across).
	126	3	} set h for the address of e, the entry point, so that any
	016	61+	} absolute fast gets planted in the right place.
	101	2*	} to read address in cue and plant
	590	v/63	} a reference to it.
	300	5	item 5 for across or down.
	590	v1/68	to complete cue action.

R 61

(Sector 34)

Build First Part of Cue.

Closed 81. Brought by R 87 into page 11 with R 60 (A, D Entries) and R 62 (U Entry). Entered at the first instruction from R 60 or R 62.

This routine builds and plants in the programme the two instructions

300 α
590 4

For an UP cue, α = 1. For an ACROSS or DOWN cue, α = 4 x, that is the address of the register 4 on from the one containing the "300" instruction.

```
Entry ──→  670  30
           680  12
           011  0+v8        plant return address
           101  v2          } make h even (only required for U; for A or D h has
           590  v4/20       } already been set to an even register, so that R 20 does nothing here).
           300  30   (2     } function "300" from function table (R 16)
           207  v/16        }
           350  -8          }
           101  v3          } plant function and step h.
           590  v/4         }
           300  1    (3     α = 1 for UP cue
           100  (-1) (1     address part set to 0 for UP by R 62, left at -1 for ACROSS or DOWN.
           090  v9          jump if UP with 1 in Sac.
           006  61+         } Sac = h for the half register 8 on from the beginning
           306  7           }   of the cue
           210  25+         }
           400  24          } set the accumulator to ½h+64p for the half register 8 on
           200  62          } from the beginning of the cue, i.e. the address 4 x which
           210  27          } is the required α for ACROSS or DOWN.
           440  26          }
           440  18          destandardise.
           410  40   (4     } 4 x as an integer in Sac.
           200  41          }
           101  v5   (9     } plant α and step h.
           590  v/4         }
           006  61+  (5     } inspect h in case the first instruction of the cue is
           101  v6          } at the end of a page (only possible in an UP cue).
           590  v3/20       }
           300  59   (6     } function "590" from function table (R 16)
           207  v/16        }
           350  -8          }
           101  v7          } plant function and step h.
           590  v/4         }
           300  4    (7     }
           101  v8          } plant "4" in address part, and step h.
           590  v/4         }
           590  (0)  (8     exit.
```

R 62.

U Entry.

```
(U) ──→  010  0+v1/61     set switch in R 61 to UP.
         101  v/18        return address for R 61 to terminate cue (no address to be read)
         590  v/61        enter R 61.
```

July 1959.

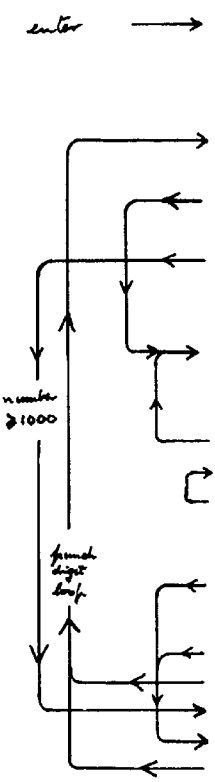
LINE 12.0

R 47.

(Sector 35)

Punch Sac.

A copy of the "Punch Sac" routine to be obeyed in page 12. Used by * printing, NOT SET printing and fault printing. Closed B1. (cf. sector 0)



015	0+v6		preserve B5
016	0+v7		preserve B6
210	0+v4		store number to be printed.
106	-30		set B6 for count of 3 digits
105	$43	(1	initial clutched count setting.
370	0		SR = 5.
290	v2		jump if number < 512
330	1000		} jump if number ≥ 1000
290	v5		
320	500		number - 500
105	404		initial clutched count setting for starting at '5'.
125	177	(2	} form clutched count version of digit.
236	5+v5		
570	100		
290	v2		
125	464	(3	} adjust clutched count.
090	v3		
226	5+v5		restore Sac to number - digits formed
126	10		step digit count
090	1v5		avoid zero suppression last time.
370	(0)	(4	compare Sac with number to be printed.
280	1v5		} back if the same, i.e. no digit yet required.
590	v1		
105	1	(5	arrange to print "1" if number ≥ 1000
625	0		punch digit.
080	v1		although at end of 3 digit count.
105	(0)	(6	restore B5
106	(0)	(7	restore B6
591	0		exit.

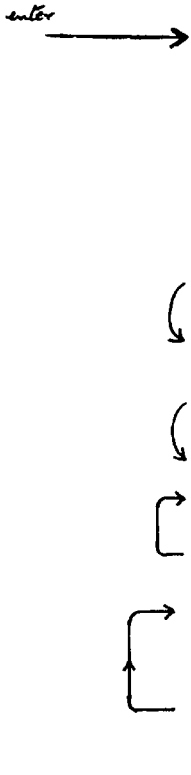
July 1959

R 46

(Sector 35)

Label Not Set Punching.

Brought down (with R 47) to page 12 by R 59.
Closed B1. Entered from R 59 with the reference to the unset label in L 44.




```
620  0          } punch ϕ CR LF
620  30         }
620  13         }
620  12           punch v
200  44 +       } label number
340  0          }   from reference.
340  0          }
340  0          }
101  v1           return address for R 47.
280  v/47         punch label number if non-zero.
620  23   (1      punch /.
200  44           routine number from reference.
101  v2         } punch routine number.
590  v/47       }
106  v5-n6  (2  }
206  v6     (3  } punch " NOT SET - "
627  0          }
186  v3         }
106  -876       }
580  0      (7  } loop
200  -15        }
280  *          }
186  v7         }
590  v5/39  (4    out 16 R 59.

= 14 (5,  = 27     Sp   λ
= 14 ,    = 15     N    0
= 20 ,    = 0      T    ϕ
= 14,     = 27     Sp   λ
= 19,     = 5      S    E
= 20,     = 0      T    ϕ
= 14,     = 11     Sp   -
= 14 (6            Sp
```

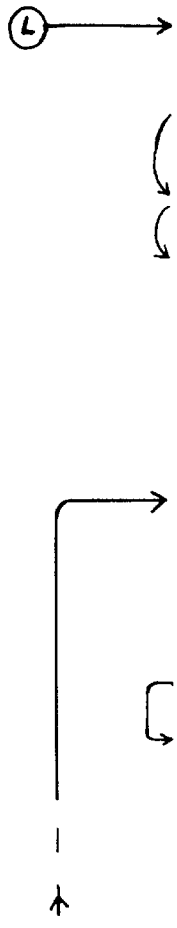
CHAPTER 6

(Sector 36)

C6 P11
R49

L Directive Entry.

Brought down to page 11 and entered at the first instruction by R 81 for L at the beginning of a line.



} plant type 7 number for L directive.

} read address in directive. Return to v1 after reading a terminating character, with the absolute part of the address (i.e. the part just read) in the accumulator and perhaps some in L 50.

} to R 32 to add the contents of L 50 to the accumulator and then to R 84 to adjust the sign.

store absolute part of the address in L 50

} accumulator and label storage space to zero.

} if the terminating character read was not CR, back to complete the necessary action, with eventual return to line 11.63 with the label of the symbolic part of the address in L 42 and the accumulator. If the terminating character was CR, through to add a zero 'label' to the absolute part of the address.

} value of label, without over- or underflow, to H 23

overflow and underflow bits to B7. Set Bt -ve for underflow

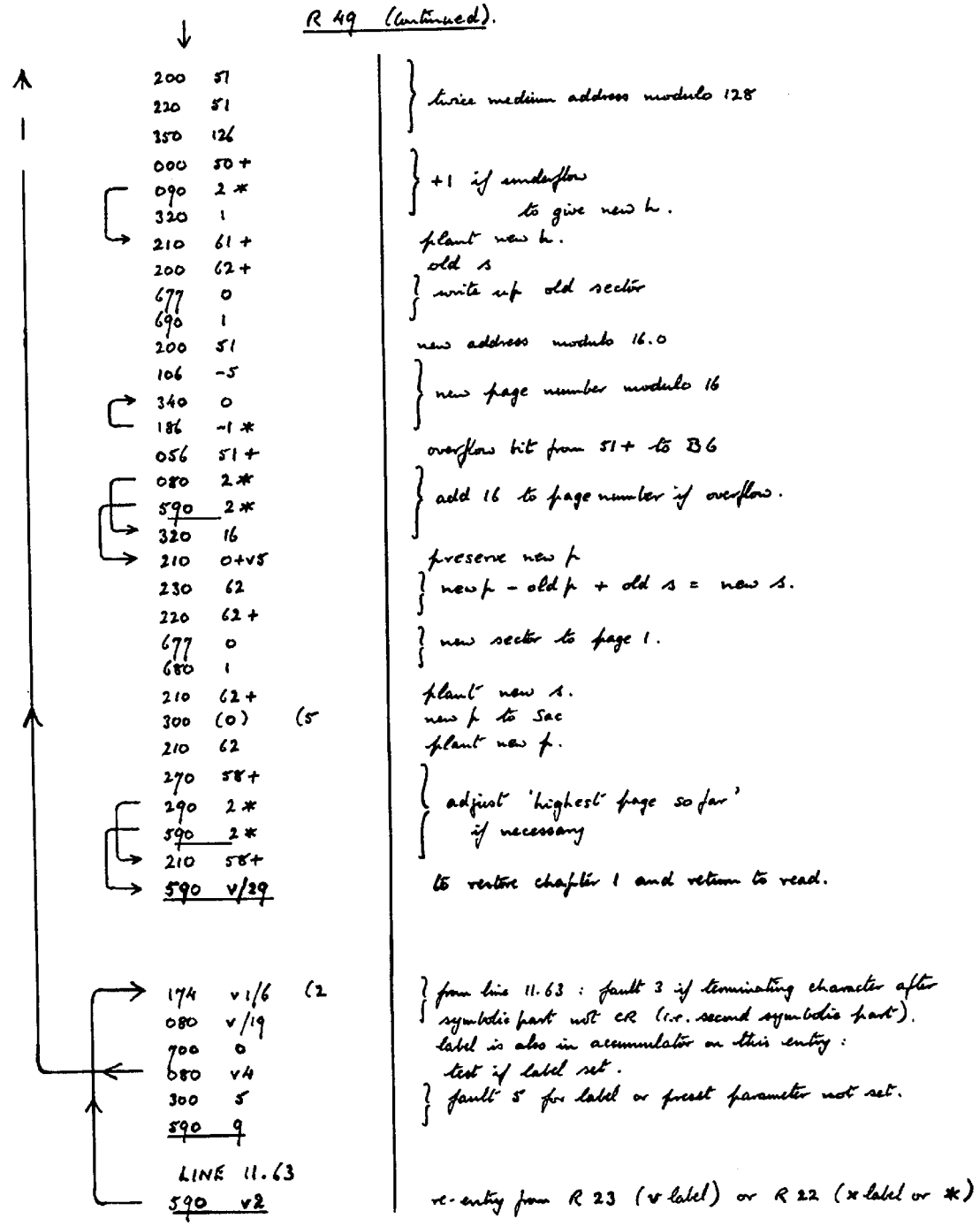
} overflow bit to H 23+

value of label, with overflow, to the accumulator.

} add ½ if underflow

add absolute part
destandardise complete address
and plant in L 50.
clear H 23+

R 49 (continued).



CHAPTER 7.

(Sector 37)

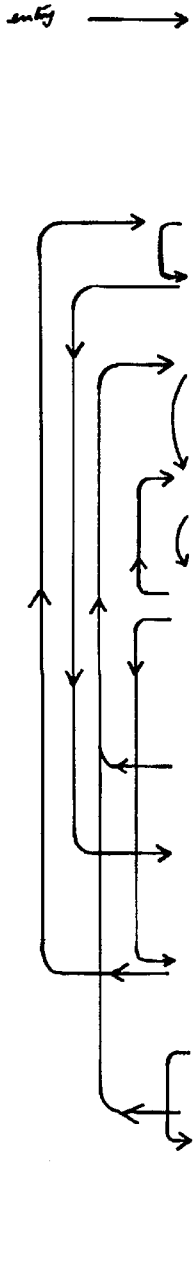
C 7 P 11

R 71

Compile n-printing sequence.

Entered from R 3 when n is read at the beginning of a line. R 71 inserts into the programme the instructions to punch the query letter, and then, except for ? 8, enters the quicky routine to insert "quicky 0" into the programme. ("quicky 0" is stored in sector 29).

For ? 8, three punch instructions are inserted into the programme. For other ? numbers, the third punch instruction ("620 0") is not inserted: instead the register number is made even by the insertion of a dummy instruction if necessary, and then a dummy instruction is inserted with the ? number in the address part, from which it is used by the n-printing routine R 72.



```
entry ──►  070  62
           090  v/19
           610  1+ *
           300  (0)
           350  16
           280  v/29
           200  56
           280  2*        (6
           590  v/29
           090  v8
           102  -4
           105  0
           101  v3        (4
           590  v4/20
           (620) (27)     (5
           205  v5        (3
           101  2*
           590  v/4
           165  1
           182  v3
           090  v7
           200  56+
           320  1
           210  56+
           210  0+v5
           590  v4

           LINE 11. 26
           010  47        (8
           670  41
           680  11

           080  v6        (7
           200  56
           370  8
           280  3*
           010  0+v5
           590  v4
           210  0+v5
           300  776
           210  v5
           101  v3
           590  v/20
```

```
BS = - page number
fault 3 for n in page 0
} ignore the n if h.s. 4 is set

? number.
} return to restore chapter 1 for ? 0 or after
    planting "620 0" instruction.
jump after inserting query number to enter R 67
set count in B2
set modifier in B5
} to make h even, i.e. set to an even half register.

instruction for planting in programme.
} plant in programme: first time : 620 27
                      second time : 620 (? letter)
                      third time : 620 0
                      or 570 (? number).
jump except after planting 620 27.
} advance query letter

plant in instruction

set quicky number zero.
} enter R 67 at line 29 of page 11.

through after planting 620 ? letter,
? number
} jump unless ? 8
set instruction to "620 0"
to plant.
set ? number.
} plant "570" function
} make register even
    and return to plant instruction.
```

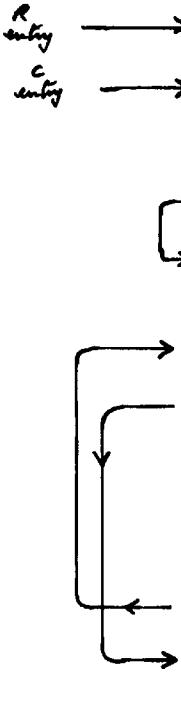
July 1959.

R 66

(Sector 37)

* Printing.

R 66 is obeyed in page 13, though appearing as it does immediately after R71 its symbolic addresses have values in page 11. It is brought to page 13 by R 55 ("Terminate C") or R 50 ("R Entry"). For 'R' punching it is entered at the first instruction, for 'C' punching it is entered at the third instruction with Sac = 3 and B6 = Bt = 0.




```
300  18
106  4
670  35
680  12
620  30
620  13
080  2.2 *
620  13
620  27
627  0
620  0
101  2.0 v3   (1
206  2.0 + v2
280  2.0 v4
590  5        (2
= 3,  = 2
= 0,  = 4
= -36,  = 0

620  14       (3
590  2.0 v1

207  61       (4
186  v/47
```

set Sac to punch 'R'
set B6 for 'R' part of table.
} punch sac routine (R47) to page 12.

punch CR
punch LF
} enter LF for 'C'

} punch C or R.

set link for R47
table look up for information to be punched.
} out to R29 if end indicated by zero table entry.
(v/29 is line 0.5, 5 is the first table entry)
table: s (62+), f (62); close, c (63+)
end , v (63)
address (43), end.

} from R47: punch s/p and re-enter punching loop.

table look-up.
enter R47, stiffning B6 (the jump is unconditional).

CHAPTER 14

(Sector 38)

PAGE 0 DURING INPUT.

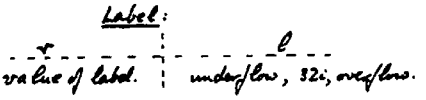
C 14 P 0
R 102

0	+0	
	-1	
	=0, =0	spare
	300 15	this line is v/29. Sets Sac to restore pages 11-15 of Chapter 1.
	330 18	this line entered from line 13 with Sac = 6 to bring down Chapter 1.
	590 52	
	300 3	this line is v/19. Fault 3 entry.
	670 127	} enter fault print,
10	690 0	preserving page 0 on sector 127.
	670 12	
	680 0	
	597 0	
14	=9, =0	} from fault print (Sac = 5) or initial entry (Sac = 6)
	=0, =0	fixed to floating conversion, range -512 to +511
16	+10	
18	=19, =0	} destandardiser for floating to fixed
	=0, =384	conversion.
20	+0.5	
22	=19, =0	} fixed to floating conversion,
	=0, =0	range 0 to +1023
24	=8, =0	} fixed to floating conversion $\times \frac{1}{8}$,
	=0, =0	range -512 to +511
26	=25, =0	} fixed to floating conversion $\times 64$,
	=0, =0	range 0 to +1023
28	=16, =0	} destandardiser for floating point to fixed
	=0, =384	conversion $\times 8$
30	+1000	
32	=0, =4	} Δ table: subtracted from h on setting a
	=2, =1	label, according to the item labelled.
	=4, =8	(see R 11)
36	=0, =32	} table of 32i for planting in label.
	=64, =96	(see R 11)
	=128, =0	
38	=0, =0	} programme zero.
	=0, =0	
	L 48	
	+0.1	
	L 52	
	677 25	} bring to the computing store
	687 14	some or all of Chapter 1.
	380 52	
	590 v1/6	enter Chapter 1.
	L 5	
	R 29	v/29 setting
	L 8	
	R 19	v/19 setting.

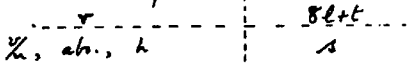
PAGE 0 DURING INPUT – WORKING SPACE.

40 Floating point working space.

42 Temporary label storage.



44 Temporary reference storage.



46 Space for destandardised parameter or quicky number.

50 Temporary storage for building up an absolute address.

56	query number	query letter	? no.	? letter
57	predicted last page	first sector	f_e	f.
58	* printing indicator: −1: no printing 0: printing	highest page so far in chapter	*	f_x
59	next position in label list	sector number of end of label list (see R51)	l_l	s_l
60	spare	next position in reference list		l_r
61	item number	half register	i	h
62	page	sector	p	s
63	routine	chapter	r	c

The initial contents of these registers (i.e. as stored on sector 38) are all zero, except for:

* printing indicator (58) set to −1.
first sector (57+)
half register (61+) } set to 128.
and sector (62+)

CHAPTER 8

(Sector 39)

C 8 P 11
R 64

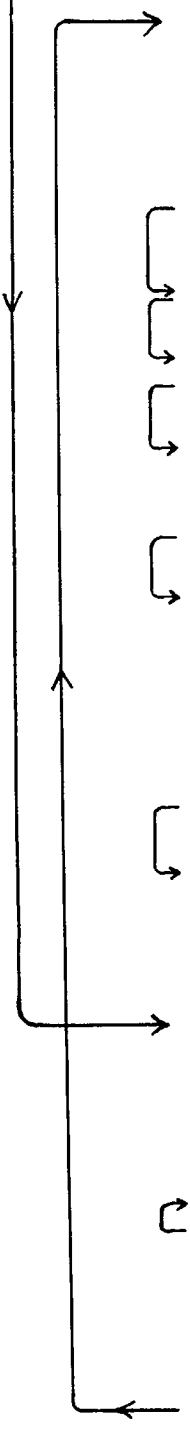
M Entry.

Brought to page 11 and entered at the second instruction from R 81.


```
680  11        bring R 53 to page 11 and enter at the second instruction.
102  v/65      set B2 for entry to R 65 after R 53.
670  26        } select R 53 and jump to enter it to terminate the
590  v         } previous chapter (or M).

101  v4   (1   } return from R 65 : to R 54 to fill in cues.
590  v/54      }

=25, =0   (6   } destandardiser for sector. line address after M.
=0, =284       }

010  63+  (4   clear e
010  63        clear r
300  7         } set type 7 for M directive.
210  0+v/8     }

102  v2        } read address in M directive.
590  v3/3      }
101  2*   (2   } after absolute part of address : adjust absolute part for sign,
590  14/32     } including any part in the L 50 working space.
410. 50        store absolute part in L 50.
174  v2/3      } test if terminating character after absolute part is v, and
080  v5        } jump if not.
400  0         } clear accumulator and return to complete v - action
594  0         } (re-enter R 64 at line 11.63).
```

July 1959.

R 64 (Continued)



```
174  v1/6  (3      } after completing label action: fault 3 if label
080  v/19          } not terminated by CR.
                   re-entry with reference in L 44 (with h = -1) and
                   label in the accumulator and L 42.
007  45            V/h indicator and abs. address indicator with h = -1 to B7.
700  0             set B1 according to first part of label.
187  3 *           jump if label set and correct B7 for h = -1.
300  5             } fault 5 if label not set.
590  9             }
090  2 *           } fault 3 if no reference in M directive.
590  v/19          }
300  0             } Sac = value of label if ordinary reference
080  2 *           }     = 0 if abs. address in reference.
200  43            }
210  23            } value of label without underflow
400  22            } to the accumulator
000  43 +          } add ½ if underflow
090  2 *           }
440  20            }
440  50            add absolute part
440  18            } destandardise and plant in L 50
410  50            }
200  51            } 2 × medium register address (modulo 128)
220  51            }
350  126           }
000  50 +          } plus 1 if underflow in total address
090  2 *           } to give h.
320  1             }
210  61 +          plant new h.
006  42            routine number to B6
101  v1/65         set B1 to return from R54 to "Terminate M" routine.
590  1v3/54        to part of R 54 to look up chapter list for s1 and p1 of
                   new chapter: enter R 65 with s1 - p1 in Sac.
174  v1/6  (5      if terminating character after address is not v, it must
080  v/19          be CR (i.e. sector.line case). Fault 3 otherwise.
440  v6            destandardise sector.line address
410  40            plant in L 40
200  40 +          line × 2^4
106  -2            } shift down 3 places to give new h.
340  0             }
186  -1 *          }
210  61 +          set new h.
300  41            new s1.
590  1v4/65        to plant s1, bring new sector to p1 and re-enter Chapter 1.

LINE 11.63
590  v3            re-entry from R 23 after reading symbolic address.
```

CHAPTER 9.

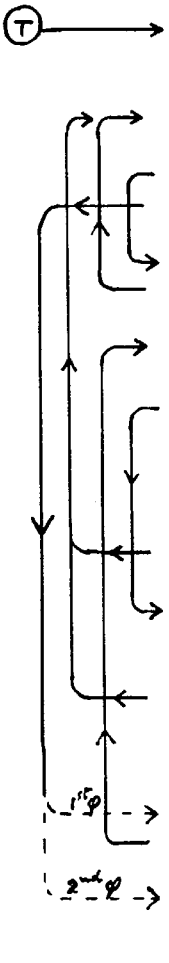
(Sector 40).

C9 P11

R48

T Entry.

This routine is brought into page 11 and entered from R 81. The title of a programme is required to appear on the programme tape before the first chapter heading, so that when the T directive is read the "first sector" is in page 1 and the current sector number is still that of the "first sector".




```
101   v7                 set B1 for first figure shift.
102   1                  set two-way switch in B2
103   0                  start word count in B3 at zero.
600   0+v2   (1          } character from tape to Sac.
300   (0)    (2
280   v3                 } jump B1 if ϕ.
591   0
370   31     (3          } ignore Erase.
290   v1
101   v7                 reset B1 for first ϕ for all characters except Er and ϕ.
627   0      (4          copy character to punch.
162   1                  } switch and test B2
080   v6
101   -4
327   0      (5          } shift alternate characters up five places.
186   v5                 Plant in next half register in page 1.
213   1.0
590   v1
223   1.0    (6          } add every other character in to least significant
213   1.0                five bits of word in page 1.
173   122                } back if not too many characters
183   v1
300   10                 } fault 10 if title is too long.
590   9
101   v8     (7          } first ϕ: set B1 for second ϕ and return to
590   v4                 copy and pack character.
013   1.61   (8          } second consecutive ϕ: plant title count
590   v29                and return to restore chapter 1.
```

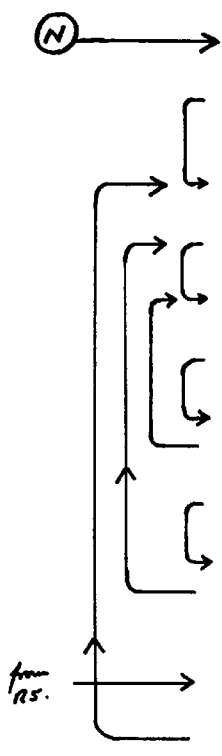
July 1959.

R 73

(Sector 40).

N Entry, and ignore characters to).

This routine is brought into page 11 and entered, from R81 for N and from R5 after a (at the beginning of a line. The N entry is at the first instruction, the (entry at v1.




```
200  58      * switch: St=0 if * set (copy N sequence)
                      =-1 if * not set (ignore N sequence).
280  3*      jump if no punching required.
620  30      punch CR
106  13      set B6 to punch LF.
101  v/29    set B1 to return to chapter 1 when terminating
              character is read.
280  v2  (3  } punch character if required (LF first time)
626  0       }
600  1+*(2   } read character to B6.
106  (0)     }
080  2*      } re-set B1 to exit address
101  v/29    }     when ) is read.
176  31      } ignore Ev.
090  v2      }
176  27      } on λ, set B1 to local terminating character
080  2*      } as any other.
101  v3      }
176  (30) (4 } test against terminating character:
080  v3      }   back to copy character to punch if not,
591  0       }   jump B1 if so.

300  6   (1  } entry from R5: set ) as terminating character
210  0+v4    }   and St ≠ 0 to prevent punching.
590  -1v3    enter read loop.
```

July 1959.

CHAPTER 15

(Section 41, 42)

C 15 P 11
R 67

Q Entry.

R 67 reads the quicky number and copies the appropriate quicky into the programme.

A quicky is inserted into a programme by being brought down from the drum into page 2 onwards of the computing store and copied into page 1 ten bits at a time. The half register count in page 1 is kept and inspected for the end of the page as in normal input.

The quickies are stored packed on the drum, so that a quicky does not necessarily start at the beginning of a sector, and may overlap more than one sector though itself less than one sector in length. The quicky table at the end of R 67 has two ten-bit entries for each quicky. The first word is the number of the first sector that contains part of the quicky. In the second word, the top two bits give the number of sectors which have to be brought down to bring the whole quicky into the computing store (at most 3), and the remaining eight bits give the half register address within the sector of the half register two before that containing the first ten bit word of the quicky programme.

With the quicky on the drum, and in fact immediately before it, are stored some indicators. The ten-bit word immediately before the first word of the quicky holds one less than the ten bit address, relative to 2.0, of the last ten bits of the quicky programme. The other indicators are to show which addresses in the quicky programme need the addition of a relativiser. For example, the addresses of all the jump instructions within a quicky will need altering according to where in the computing store the quicky is to be obeyed. The quicky is stored on the drum with all such addresses expressed relative to its first instruction, and if 'a' is the medium address of the first register in the computing store occupied by the quicky, certain quicky instructions may need $\frac{1}{2}a$, a or 2a adding to their addresses, according to their types.

For each instruction, an indicator of two bits is kept with the interpretation:

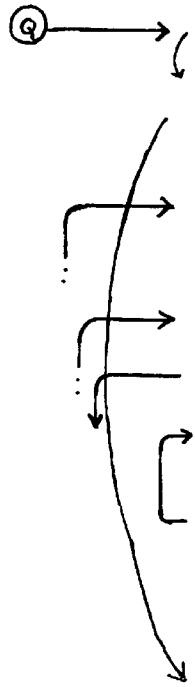
00 : no change to the address
01 : add a to the address
10 : add $\frac{1}{2}a$ to the address
11 : add 2a to the address.

These indicators are stored five to a ten bit word from the least significant end upwards, in successive ten bit words working backwards from the word before that holding the quicky's last register indicator. The address stored in the quicky table at the end of R 67 is therefore the ten-bit address, relative to 2.0, of the first of these indicator words.

In R 67, the ten-bit words of the quicky are transferred in pairs, and after each pair the appropriate two-bit indicator is inspected. B5 is used as a modifier through the words of the quicky, B4 as a modifier through the indicator words, and B2 as a counter for the five indicators in a word.

R 67 occupies page 11 and most of page 12. The first part is brought from sector 41 to page 11 by R 81 and entered at the first instruction.

R 67.




```
101  2*
590  v/20
010  61
102  v13
590  v3/3

204  2.0   (14
340  0
340  0
214  2.0
125  2     (8
175  (0)   (9
090  v3
105  1
675  12    (11
685  1
175  (0)   (10
185  v11
101  v20/51
100  (0)   (16
080  v4/20
590  v/29

440  18    (13
410  46
300  v1
210  0+v/6
174  v1/6
084  0

300  v1/6  (1
210  0+v/6

from R71.
R51 ->
670  42    (18
680  12
101  v15
200  61+
106  4
590  v1/11
006  47    (15
176  29
090  v17
026  47
206  v12
280  3*
300  11    (17
590  9
210  0+v2
206  0+v12
350  -256
105  0
675  (0)   (2
685  2
137  256
185  v2
015  0+v10
```

make register even.

clear 0 for quicky.
to read quicky number.

after planting instruction of quicky in the programme: shift indicated word down two bits.

step count through quicky.
test for end of quicky.

after planting the quicky, restore the pages of Chapter 1 overwritten by the quicky sectors.

0+v16 is set non-zero by R 51 : if R 67 was entered from R 51, re-enter R 51 at v 20 after inspecting it.

to restore chapter 1 if in Q.

quicky number to H 47.

change R 6 so that CR, after terminating the previous phrase, will return to R 67 instead of to 'read'.

if the terminating character at the end of the quicky number was not CR, return to complete the necessary action (e.g. 'addled quicky'). CR will eventually produce a return to v1.

restore R 6

rest of R 67 to page 12 : entry here from R 71 for 'quicky 0' and from R 51

to R 11 to form in L 42 a label referring to the current address.

quicky number to B6.

to give fault 11 if ≥ 29

first sector of quicky from table

fault 11 for unassigned quicky.

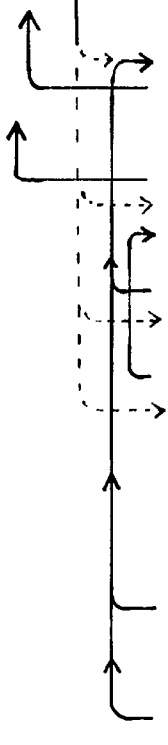
plant first sector of quicky in the sector select instruction.
second entry in table
number of sectors occupied by quicky × 256

bring down sectors containing the quicky to page 2 onwards.

plant number of sectors brought down.

July 1959.

R 67 (Continued).



```
206  0+v12        } address of indicators to Sac.
350  255          }
210  2+*          } and to B4, B5
004  1+*          }
105  (0)
102  1            set B2 for count through the indicators in one word.
207  2.0+         'end of quicky' address to Sac
210  0+v9         plant in comparison instruction.

006  61+   (3     } inspect h for end of a page.
101  2*           }
590  v3/20
205  2.1          function part of quicky instruction to Sac.
101  2*           } plant in page 1.
590  v/4
205  2.1+         address part of quicky instruction to Sac.
101  2*           } plant in page 1
590  v/4          (new h left in B6 as well as in H 61+)
204  2.0          } two bit indicator to Sac.
350  3
207  2*           } jump according to table.
597  0

=v4, = v5
=v6, = v7
172  5     (4     } test count of 5 indicators per word.
182  v14
102  1            reset count in B2
134  1            next indicator word back
590  v8

200  43    (5     add a:    a from label
226  63+                    } add to address just planted.
216  63+
590  v4

200  43    (6     add ½a:   a to Sac
340  0                      halve
590  1v5                    to add in.

206  63+   (7     add 2a:   address to Sac.
340  0                      } add a to half address as stored and set B1,
027  43                     i.e. B1 set to overflow bit of "address + 2a"
206  63+                    }
220  43                     } add 2a to address
220  43                     }
216  63+
090  v4                     }
206  63                     } if the addition of 2a has produced overflow,
320  512                    } alter the most significant function bit, i.e.
216  63                     } set the function to refer to the correct
590  v4                     } quarter of the store.
```

		Quicky table
=29(12,	= 360	n-printing
=49,	= 269	Reciprocal
=49,	= 581	Reciprocal square root
=0,	= 0	
=50,	= 274	Exponential
=50,	= 637	Tangent
=51,	= 638	Sine or cosine
=52,	= 597	Cosine
=53,	= 295	Print Sac signed
=53,	= 888	Print accumulator fixed point
=55,	= 558	Print accumulator floating point
=56,	= 642	Print Sac unsigned
=57,	= 590	Square root
=0,	= 0	
=58,	= 286	Logarithm
=58,	= 900	Arctangent
=60,	= 547	Arcsine
=0,	= 0	
=61,	= 266	Read integer to Sac
=47,	= 791	Read number to accumulator

CHAPTER 27.

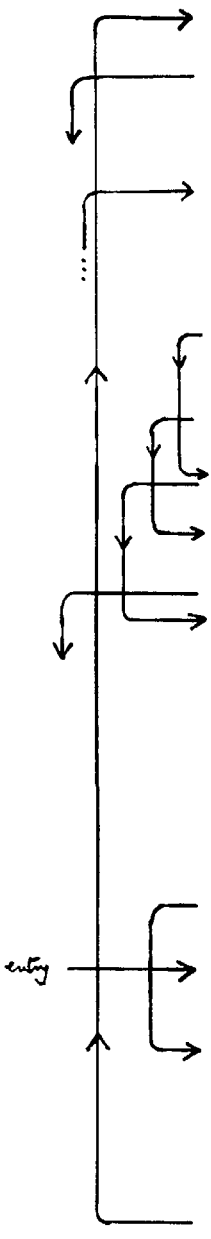
(Sector 43)

C 27 P0

R 72

First n-printing sector.

This sector is brought to page 0 by the n-printing quicky, 'quicky 0'. It is entered at v1 with the return address in Sac, quicky 0 having stored Sac in the address part of its first instruction, which was '670 479'. R 72 uses this instruction and the preceding one, '570 ?number', as working space, and restores them before exit.



```
410  1.0      complete the exchange of 1.0 and first 40 bits of 'quicky 0'.
400  v1       restore the accumulator
290  v10      jump for entry exchange, through for exit' change back.
620  30       } punch CR LF before exit
620  13       }
670  479      select sector with stored page 0.
300  (0)  (5   restore return address to Sac.
680  0        restore page 0 and run on to a '597 0' instruction.

210  0+v5     from sector 44 : store Sac.
200  1.1      printing indicator to Sac.
380  v2       step indicator ; jump except after punching return address.
200  1.0+     ? number
370  520      } jump if ? 1-7
290  1v2      }
207  -5v9     pick up indicator from table for ? 9 or 10
290  v3   (2  jump if indicator positive (i.e. after printing one B-register,
                one only required)
210  1.1      plant indicator
330  521      } through after printing the accumulator ; jump with
380  v4       }    Sac = indicator - 520
000  1.0  (3  reset B1
300  776      } restore '570' function to 1.0
210  1.0      }
300  32       }
310  1.1      } restore '670 479' to 1.1 , 1.1+.
300  479      }
210  1.1+     }
200  0+v5     restore Sac (return address)
340  0        long address
377  1        S1 negative as indicator
590  v6.      to exchange back 1.0 and 40 bits of 'quicky 0'.

entry
620  0    (1  entry ; punch 'f'
210  0+v5     preserve Sac (return address)
340  0        form long address
410  v1   (6  store the accumulator
407  -8       }
410  2v1      } exchange L 1.0 with the 40 bits at the
400  1.0      }   beginning of quicky 0 consisting of:
417  -8       }        570  ? number
400  2v1      }        670   479 ;
590  0        } and the last ten bits have been over-written by the
                stored contents of Sac.
```

July 1959.

R 72 (Continued).

```
                300  10   (10     to store sign of Bt; set Sac for '+'
                080  2*             set Sac for '0'
                300  0
                090  2*             set Sac for '-'
                300  -5
                210  1.0            store sign of Bt in 1.0  (over '570')
                627  16             punch sign of Bt.
                010  1.1            clear indicator in 1.1
to punch        107  13             set Sac for punching "M" and Bt ≠ 0
B or A          590  2v4
                100  0    (4        Bt = 0 except for "M"
                380  v7             jump except before punching accumulator: through with Sac = 1.
                210  0+v8           plant M or A in punch instruction
                620  30   (7        punch CR
                620  13             punch LF
                620  27
                620  (2)  (8        punch M, B or A
                620  0
                670  44             select punching sector
                590  v5

                = 520 , = 513 (9    table for ? numbers 9 and 10
```

Instructions inserted in the programme for n-printing:

```
                620  27
                620  ? letter

(even register:)570  ? number
                670  (479)
                690  0
                670  43
                680  0
                210  0+      + 2a
                210  8+      + 2a
                300  8       + a
                590  v1/72
                300  (0)
```

'quicky 0'

July 1959.

CHAPTER 77.

(Sector 44)

C 77 PO

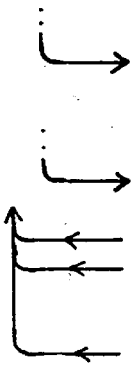
R 722

Second n-punching sector.

```
106  (0)  (6      restore B6
620  15           punch ','
186  v7           punch four parts of the accumulator.
105  (0)  (1      } restore B5, B6, Sac
106  (0)          }
300  (0)          }
670  43           } re-enter sector 43
680  0            }

entry
015  0+v1  (5     } from sector 43,
016  1+v1         }   preserve B5, B6, Sac.
210  2+v1         }
080  v20          jump straight to punch Sac for M.
200  1.1          } indicator without tip hit.
350  511          }
370  8            } punch B number except for accumulator.
290  2*           }
637  v8           }
620  10           punch =
290  v3           jump for accumulator punching.
320  200          } plant '30b' function
210  v4           }
200  1.1+         restore Sac to value before entry.
(300)  0   (4     contents of B-register to Sac.
210  0+v22 (20    }
106  -30          }
105  543          }
290  v23          }
330  1000         }
290  v24          }
320  500          }
105  404          }
125  177  (23     }
236  5+v24        }
570  100          } punch Sac routine.
290  v23          }
125  464  (26     }
090  v26          }
226  5+v24        }
126  10           }
090  1v24         }
177  0    (22     }
080  1v24         }
590  2v20         }
105  1    (24     }
625  0            }
080  2v20         }
590  (v1) (13     }
```

July 1959

R 722 (Continued).



```
106   -3   (3
010   0+v13
410   v5
100   0
206   1+v5  (7
016   0+v6
080   v20
290   v20
167   -1
620   11
380   v20
```

to punch accumulator : set B6 = -3.
set jump address at the end of the 'punch sec' to 0.0
plant the accumulator.
Bt = 0 first time
part of the accumulator
preserve B6
} punch first and last ten bits signed.

```
=16 (8, =1
= 2,  = 19
= 4,  = 21
=22,  = 7
```

} punching table for B number.

July 1959

CHAPTER 30.

(Sector 45).

Interlude.

An interlude is headed by an I directive, and terminated by a J. The interlude programme is stored on sectors 114 - 126. Subsequent J directives re-enter the interlude. The effect of a J directive is to bring sectors 114 - 126 to pages 1 - 13 of the computing store and transfer control to line 1.0.

R 666 is mostly obeyed in page 11, whither it is brought by R 81 when J is read. However, it also contains the "return to input" sequence which is left at the start of page 15 by a J directive before entering an interlude.

While the interlude programme is being read the chapter number is set to the "impossible" value of 127 as an indicator.

C 30 P 11 - 11

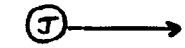
R 666

J Entry and "Return to Input".

		Return to input (obeyed in page 15):
670	127	} restore the input routine's page 0
680	0	
200	62 +	} restore "current sector" before the interlude
677	0	to page 1.
680	1	
300	6	} to bring chapter 1 to the computing store
590	6	and return to read.

R 666 (continued).

J Entry.




```
200  62+ (1      } write up current sector
677  0           }
690  1           }
670  28          } fast of R51 to page 13
680  13          }
006  63+         } jump if chapter number is not 127, i.e. if this J does not
176  127         } terminate an interlude
080  v3          } old page 0 (from when 'J' was read)
670  127         }    to page 14
680  14          }
200  14.60+      } plant in R51 the old reference list indicator, so that
210  0+v17/51    } R51 will ignore quicky references outside the interlude.
101  45   (3     this sector number in B1
010  0+v16/51    set switch to exit from R51 without packing labels.
590  v4/51       enter R51.

670  45   (2     } from line 11.63 on exit from R51: this sector
680  15          } to page 15.
006  63+         } through with B6 = 0 if J terminates an interlude
136  127         }
080  v4

105  -1          set B5 for copying loop and B1 < 0
200  58+         highest page read during interlude to Sac
370  526         } test against 14. If less, enter copying loop
290  2v7         } so that fx is copied.
300  8           } fault 8 if interlude runs into page 14.
590  9

126  (29) (7     B6 = 29 first time,          } reset fx, h, fx, s and c to the values
016  0+v7        doubled each time.          } they had before the interlude. Leave all
205  14.59                                   } the rest. B6 holds a word whose bits
090  2*                                      } indicate which numbers are to be
215  59                                      } copied from the 'old page 0' in page 14
185  v7          test on B6, step B5         } to the current page 0.

670  127  (4     } write up the new page 0.
690  0           }
670  2           } "standard page 0" to page 0.
680  0           }
590  4.1 *       to obey the rest of the routine in page 15.

300  -12         } bring sectors 114 - 126 to pages 1 - 13
677  126  (5     }
687  13          }
380  4.0 v5      }
590  1.0         enter the interlude at line 1.0
```

from 11.63

R 52

(Sector 45)

F Entry.

Brought to page 11 by R 81 and entered at the first instruction.

000	63+	} fault 3 if F is read once a chapter has been set.
080	v/19	
102	v1	} to read F number.
590	v3/3	
440	18 (1	} } f to Sac.
410	40	
200	41	
210	57+	} set first sector and current sector to f
210	62+	
677	0	} } bring to page 1 to receive title.
680	1	
590	v/29	to restore Chapter 1.

LINE 11.63

590 v2/666 return from R 51 on J entry.

CHAPTER 28.

(Sector 46)

C 28 P 15

R 81

Letter-shift Character Action

Brought down to page 15 and entered at the first instruction by R1, when letter-shift is read. In the same sector is R 45, a table of sector numbers and entry points for directive action.

```
600  0+v3
106  (0)  (3
080  3*
185  v6   (2
590  v/19

176  31
090  v

600  1+v/1
000  1+v/1
080  -2*

172  v/1
080  v1

206  16 v/45  (9
677  0
680  11

205  0+v11
080  2*
206  v/45
597  0 .....

=4,  =7   (5
```

read next character after λ to B6

jump unless φ: through to FAULT 3 if φ immediately follows λ.

cycle table to test if letter is L, P or S. Through to FAULT 3 if not, and for φ after λ.

ignore erase after λ.

after reading first letter of directive, ignore all other characters to φ.

jump to test for L, P or S if not at the beginning of a line.

Letter at the beginning of a line. Re-enter here also from R 80 for P or S in chapter heading. Required sector from table.

bring directive sector to page 11.

special entry-point look-up for P or S in chapter heading.
jump for P or S in chapter heading.
look up entry point in table.
enter.

Comparison table for L, S, P. The zero is the entry for L. P; S. The 'S' entry is also the B2-comparison entry for a letter after P in a chapter heading.

Comparison table for B2 test. ("after P" above)

=7 v2/58 w/57 , = 7 v/55 w/57	after S; after C
=7 v/31 w/57 , = 7 v/34 w/57	after =; after present parameter

(if the letter is L, only these last two are tested)

```
306  -12   (1
101  -4
280  2*
101  -1
105  -3
275  0+v5  (6
280  v2

302  7w/57
590  v1/80

=v/56, =v/58  (11
```

value of letter -12 to Sac. Sac = 0 if L.
set B1 to test setting of B2 against full table.
if the letter is L, reset B1 to test setting of B2 against the last two entries in the table only.
set B5 for loop to test for L, P and S.
test Sac for L, P or S. Leave with B5 = -3, -2, -1 for L, P and S respectively, and B6 = 0. If the letter is not L, P or S, Sac is also tested against 7v2/58w/57 when B5 = 0; however, this test must always fail since Sac must be < 19.
set Sac from B2 for test against second table, and enter R80 with B6 ≠ 0 to test if L, P or S is O.K.

special entry points for P and S respectively in chapter heading.

July 1959

R 45

(Sector 46).

Letter-shift Table.

Used by R 81.

= v/19	= v/60	Entry points	
= v/19	= 1v/53	ϕ	A
= v1/60	= v/53	B	C
= v/52	= v/19	D	E
= v/19	= v/667	F	G
= v1/666	= v/19	H	I
= v/49	= 1v/64	J	K
= v/73	= v/19	L	M
= v/19	= v/67	N	O
= v/50	= v/19	P	Q
= v/48	= v/62	R	S
= v/19	= v/10	T	U
= v/19	= v/19	V	W
= v/19	= v/19	X	Y
= v/19	= v/70	Z	λ
= v/19	= v/19	.	?
		£	Ev

		Sectors	
= 22	= 34	ϕ	A
= 22	= 26	B	C
= 34	= 26	D	E
= 45	= 22	F	G
= 22	= 32	H	I
= 45	= 22	J	K
= 36	= 39	L	M
= 40	= 22	N	O
= 26	= 41	P	Q
= 31	= 26	R	S
= 40	= 34	T	U
= 22	= 22	V	W
= 22	= 22	X	Y
= 22	= 22	Z	λ
= 21	= 29	.	?
= 22	= 22	£	Ev

QUICKY ROUTINES.

Quicky 1 : $A' = 1/A$

To find the reciprocal of $A = a.2^p$

707	-259		To error print if $p < -253$ or $p > 259$
300	1		
← 090	9		
210	2		H2 = 1 : L2 = -2
410	32		Store A
410	34		Form L34 = $b.2^q$ as a first approximation to $1/(a.2^p)$
230	32		$q = 1 - p$
210	34		
300	749		$b = 2(\sqrt{3}-1) - a, \; a \geq 0$ $= -2(\sqrt{3}-1) - a, \; a < 0$
230	33+		
490	v1		
330	475		
210	35+	(1	
300	-1		$y_{n+1} = y_n(2 - x y_n)$
510	34		
430	2	(2	
500	34		
410	34		
510	32		
380	v2		
010	2		Restore L2 = -1
450	2		Final iteration written out for accuracy
500	34		
420	34		

September 1960.

Quicky 2: $A' = 1/\sqrt{A}$

To find the reciprocal square root of $A = a.2^b$

```
410  32          Store a.2^b
490  v1
300  2    (6     To error print if a.2^b < 0
590  9
=1,  =0   (2     -1·5
=0,  =-384
=0,  =0          -0·5
=0,  =-256
410  34   (1
707  512         H34 = q = 1 - [(b+1)/2]
347  -257
210  34
707  -1          H32 = b-1
210  32
157  1           Bt = 1 (b even); = 0 (b odd)      Form in L34 a first
230  33+                                          approximation
290  v6          To error print if A = 0
080  v3
320  247         H35+ = b = 512 (0·975 - a/2), b even
340  13   (3                 = 512 (1·216 - a/2), b odd
210  35+
400  34
300  -2
590  v5
430  v2   (4
500  34
410  34          y_{n+1} = y_n (3/2 - x y_n^2 /2)
510  32   (5
500  34
380  v4
430  2v2
500  34          Final iteration written out for accuracy
420  34
```

September 1960.

Quicky 4: $A' = e^A$

```
500  2v3      (Store A/(2 log_e 2)
410  32
450  v3       (To error print if A/(2 log_e 2) ≥ 127·5
300  4
490  9
400  32
440  v3       (A' = A/(2 log_e 2) + 128·5
450  2
300  -5       Set count for power series
490  v7
400  0        (Exit with A'=0 if e^A < 2^-257
590  1v9
=8,  =0   (3  (+127·5
=0,  =255
=0,  =869     (1/(2 log_e 2)
=337, =369
=-8, =532 (4  (-2·6565677, -3
=817, =-349   (-9·3553891, -3
=-6, =238     (+4·2833310, -2
=453, =-307   (-1·5386563, -1   } Coefficients in
=-4, =863     (+4·4403370, -1   } power series
=911, =350    (-9·6090614, -1
=-2, =399     (+1·3862944
=904, =-316
=-1, =84   (5  (Destandardiser
=707, =454
=0, =454      (H34+ = [A/(log_e 2)]
=16, =-492    (L32 = ½[A/(log_e 2)] - A/(2 log_e 2) = x
=1, =766   (6
=912, =354    (Sum power series for 2^-x
=28, =0    (7
=0, =640
400  32   (7
440  v6
410  34
450  v6
450  32
410  32
400  v4
500  32   (8
437  v5
380  v8
500  32
450  2
900  0        (Adjust exponent
220  34+
717  0    (9
```

September 1960.

Quicky 5: $A' = \tan A$

```
500  v5
410  32
450  2
490  4*
900  1
290  2*
400  2
440  2v6
410  34
450  2v6
450  32
410  32
500  32
410  36
400  v2
300  ≠2v2n3
500  36   (1
427  v3
380  v1
510  32
250  34+
410  32
280  v7
590  1v10
=-13, =219
=798, =496   (2
=-18, =370
=683, =-286
=-10, =513
=78, =403
=-8, =187
=365, =316
=-6, =442
=651, =326
=-4, =130
=410, =326
=-2, =492
=751, =330
=0, =852
=126, =402   (3
=1, =110
=972, =325   (5
```

$L32 = 4A/\pi$

$A' = \frac{4A}{\pi} + 1$

Ensure that small negative number is destandardised correctly

$H34 += \left[\frac{2A}{\pi} + \frac{1}{2}\right]$

$L32 = x = 1 - 2\,\mathrm{Frac}\left[\frac{2A}{\pi} + \frac{1}{2}\right]$

Sum power series for $\tan \frac{\pi}{4} x$

$$= \begin{cases} \tan A & \text{if } \frac{4r-1}{4}\pi \leq x < \frac{4r+1}{4}\pi \\ -\cot A & \text{if } \frac{4r+1}{4}\pi \leq x < \frac{4r+3}{4}\pi \end{cases}$$

Jump to form $\tan A$ if $A' = -\cot A$

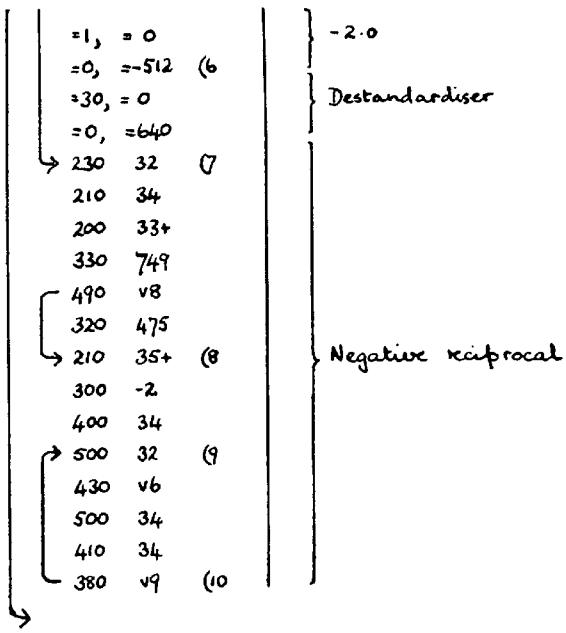
Coefficients in power series:

+1·1844146, -4
-2·1258939, -6
+7·6880772, -4
+2·4136095, -3
+9·9681446, -3
+3·9843813, -2
+1·6149115, -1
+7·8539816, -1

$4/\pi$

September 1960.

Quicky 5: (continued)



September 1960.

Quicky 6 : A' = sin A or cosA

```
590  v1
420  v5        A' = A + π/2
500  v7  (1
410  32        Store 2A/π
420  v6        H34 + = [2A/π] + 1
410  34
450  v6
450  32        A' = [2A/π] − 2A/π = x : −1 < x ≤ 0
410  32
500  32
410  36
500  v3
300  -3        Sum series for sin(π/2)x / x
427  v4  (2
500  36
380  v2
420  v5
200  34+
350  2         Arrange for adjusting sign if necessary
280  v8
510  32        A' = −Ax
590  1v8
=-18, =451  (3    -3·4592635, -6
=722, =-465
=-12, =245        +1·6031984, -4
=220, =336
=-7,  =714        -4·6817109, -3   } Coefficients in power series
=183, =-307
=-3,  =80         +7·9692621, -2
=431, =326
=0,   =794  (4    -6·4596410, -1
=272, =-331
=1,   =852  (5    π/2
=126, =402
=29,  =1    (6    Destandardiser
=0,   =640
=0,   =110  (7    2/π
=972, =325
500  32  (8       A' = Ax
```

September 1960.

Quicky 7 : $A' = \cos A$

```
500  v7   (1        A' = 2A/π
410  32              Store 2A/π
440  v6              H34+ = [A/π]
410  34
450  v6              L32 = A' = 1 + [2A/π] - 2A/π = x :  0 < x ≤ 1
450  2
450  32
410  32
500  32              Sum series for sin(π/2)x / x
410  36
500  v3
300  -3
427  v4   (2
500  36
380  v2
420  v5              Arrange for adjusting sign if necessary
250  34+
280  2v7             A' = Ax
500  32
590  3v7
=-18, =519  (3       -3.3830983, -6      Coefficients in
=950, =-455          +1.6014920, -4      power series
=-12, =812           -4.6815920, -3
=877, =335           +7.9692595, -2
=7,   =694           -6.4596409, -1
=191, =-307
=-3,  =992
=430, =326
=0,   =794  (4       π/2
=272, =-331
=1,   =852  (5       Destandardiser
=126, =402
=30,  =0    (6       2/π
=0,   =640
=0,   =110  (7
=972, =325
510  32              A' = -Ax
```

September 1960.

Quicksy 8: Print Sac (signed integer)

```
620  0
620  30          } Punch FS CR LF CR
620  13
620  30
015  0+v7        } Store B5, B6
016  0+v8
370  0           Ensure St = S
290  v1
620  11
167  -1          } Punch sp or - ; form S' = |S|
380  1v1
620  14   (1
106  -30         Set digit count in B6
210  0+v5        Store Sac for zero-suppression
105  542  (2
125  177  (3
570  100
236  5+v6        } Form digit to be punched
290  v3          using clutched count in B5.
125  464  (4
090  v4
126  10          Count digits
226  0+v6        Restore S ≥ 0
090  v6          Jump on last digit
370  0̲   (5      } Suppress non-significant zeros.
290  v2
625  1    (6     } Punch digits
080  v2
105  0̲   (7      } Restore B5, B6
106  0̲   (8
620  14          } Punch two spaces.
620  14
```

September 1960.

Quicky 9: Print Accumulator Fixed Point

```
620  0          }
620  30         } Punch FS CR LF CR
620  13         }
620  30         }
016  0+v9         Store B6
011  34         }
006  34         } B6 = 281
026  34         }
206  8·2        }
210  0+v5       }
206  8·2+       }
171  4·0        } H(0+v5) = m
090  4*         } H(0+v1) = n
206  0·2        }
210  0+v5       }
206  0·2+       }
210  0+v1       }
337  0          (1   S = -n
410  32           Store A
400  2v3        }
590  2*         } L34 = round-off constant
500  v3         }
380  -1*        }
410  34         }
440  32           Round A
106  14         }
490  5*         }
400  32         }
520  2          } L32 = |A|r ; set sign in B6
440  34         }
106  11         }
410  32         }
400  v2           A' = 10
590  3v4        }
+10         (2  }
=-3, =410   (3  } +0·1
=614,=409       }
=0,  =0         } +0·5
=0,  =256       }
320  1      (4  }
400  34         }
520  v2         }
410  34         } L34 = 10^m', m' such that |A| < 10^m'
400  32         }
450  34         }
490  v4         }
```

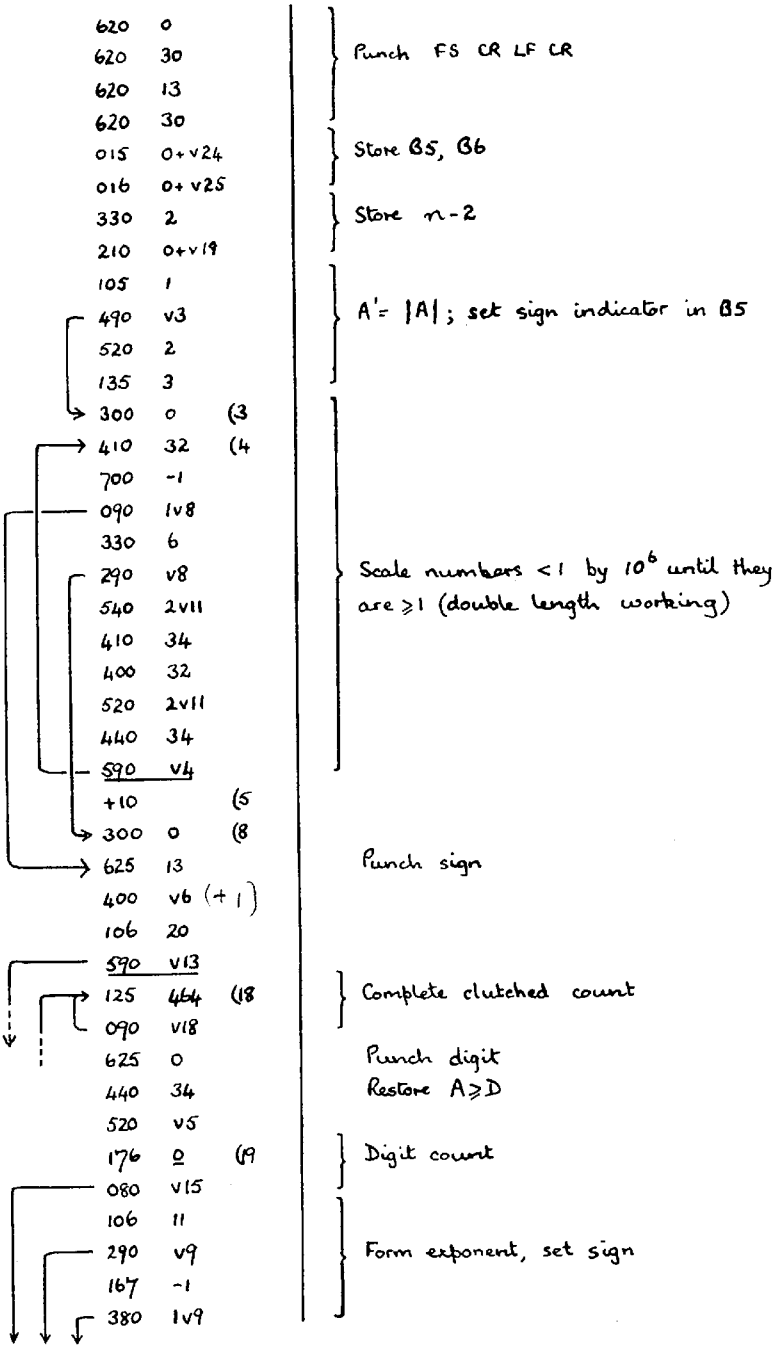
September 1960.

Quicky 9 (continued)

```
330  0      (5    S' = m' - m
290  4*           Punch spaces for non-significant digits
320  1
620  14
380  -1*
626  0            Punch sign (space or -)
106  0            Clear sign
400  34     (6    Multiply subtraction constant by 10^-1
520  v3
410  34
400  32
300  543    (7    Calculate and print decimal
410  32           digit using clutched count
450  34
320  177
490  1v7
320  464
290  -1*
627  0
080  -4v9   (8
200  34           Cycle for digits before decimal point
370  2
290  v6
036  0+v1         Test for n = 0
090  v9
620  28           Punch decimal point
400  32           Cycle for digits after decimal point
520  v2
010  0+v1
186  v7
106  0      (9    Restore B6
620  14           Punch 2 spaces
620  14
```

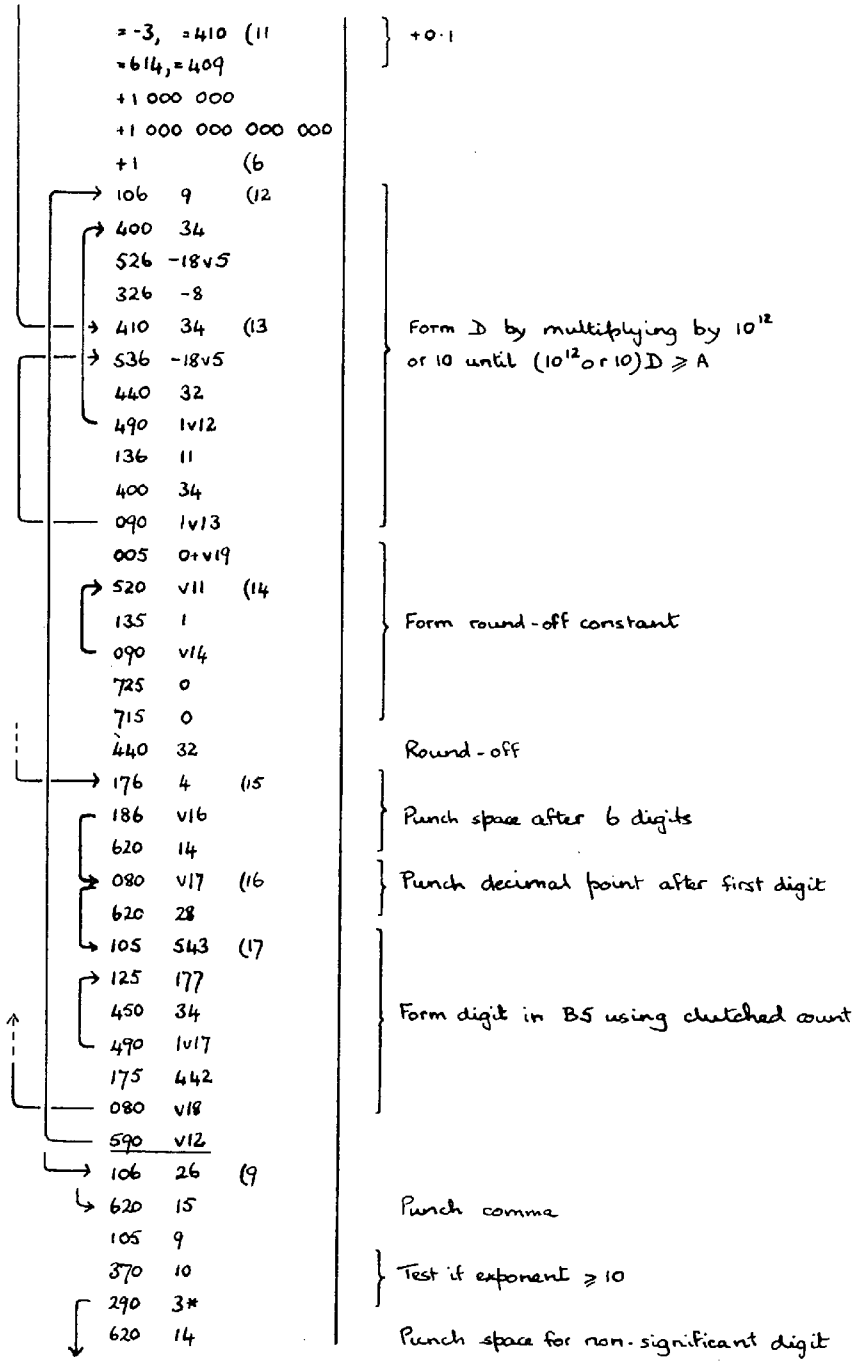
September 1960.

Quicky 10: Print Accumulator Floating Point



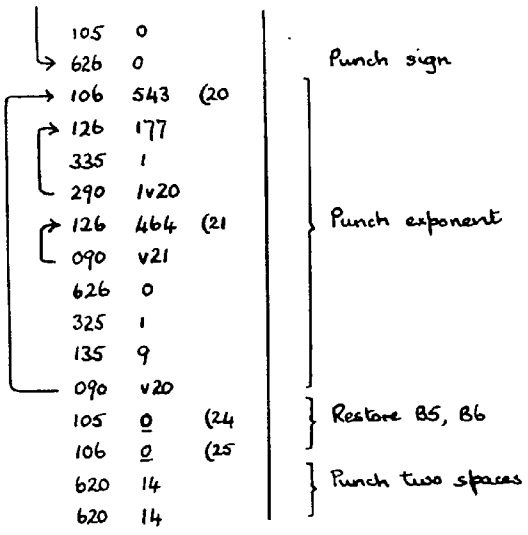
September 1960.

Quicky 10 (continued)

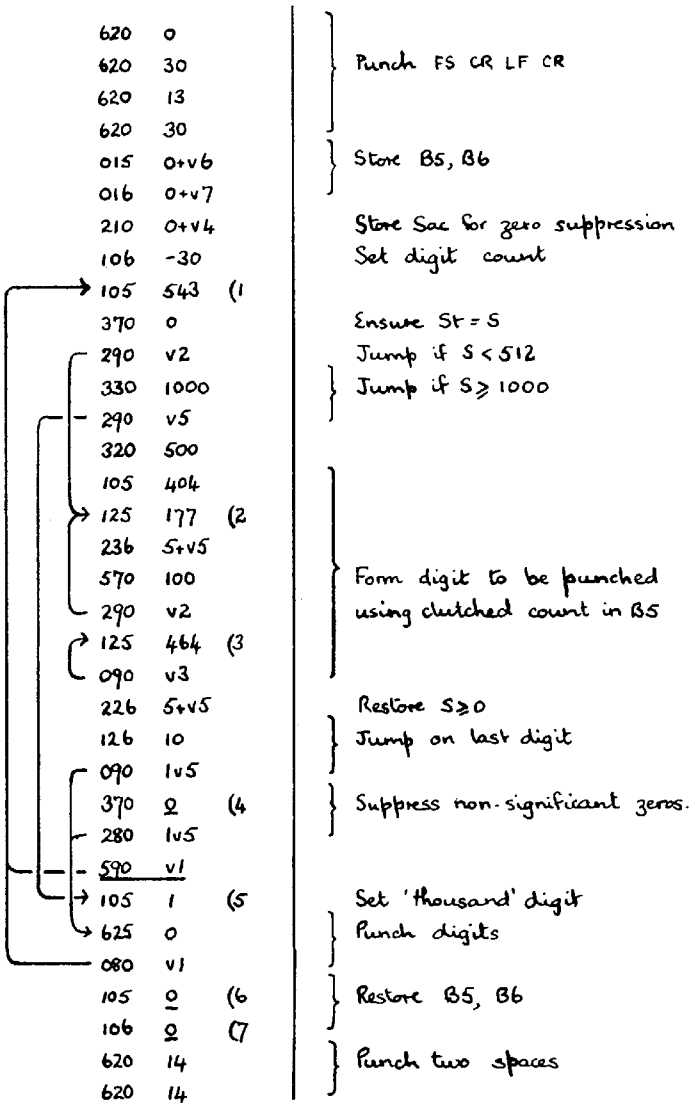


September 1960.

Quicky 10 (continued)



Quicky 11: Print Sac (unsigned integer)



September 1960.

Quicky 12 : $A' = \sqrt{A}$

	410 32		Store A
	490 v1		To error print if A<0
	300 12		
←	590 9		
	=1, =0		-1·5
	=0, =-384	(2	
	=-1, =0		-0·5
	=0, =-512		
→	410 34	(1	Store A
	410 36		
	707 512		$q = 1 - \left[\frac{p+1}{2}\right]$
	347 -257		
	210 34		L32 = ½ A
	707 -1		
	210 32		
	157 1		Bt = 1 (p even) ; = 0 (p odd)
	230 33+		
	290 1v6		Exit with A' = 0 if A = 0
	080 v3		$b = 512(0.975 - a/2)$, p even
	320 247		$= 512(1.216 - a/2)$, p odd
→	340 13	(3	
	210 35+		
	400 34		
	300 -2		
	590 v5		
→	430 v2	(4	$y_{n+1} = y_n(3/2 - x y_n^2/2)$
	500 34		
	410 34		
→	510 32	(5	
	500 34		
	380 v4		
	430 2v2		Final iteration written out for accuracy
	500 34		
	420 34		
	500 36	(6	$A' = \frac{1}{\sqrt{A}} \times A = \sqrt{A}$

September 1960.

Quicky 14 : $A' = \log_e A$.

```
900  0
370  257
290  2*
490  v6
300  14
590  9
=0,  =767
=912, =354  (2
=9,  =0
=0,  =0     (1
=1016,=680
=221, =498  (3
=1019,=381
=841, =642
=1021,=162
=246, =274
=1022,=263
=528, =766
=1022,=688
=823, =378
=1022,=776
=272, =519
=1023,=632
=844, =340
=1023,=777
=37,  =512
=0,   =476
=1023,=511  (4
210  v1   (6
300  1
717  0
440  2
410  32
500  v3
300  #2v3n4
427  v4   (7
500  32
380  v7
410  32
400  v1
440  2
500  v2
420  32
```

To error print if $A<0$ or exponent $p>256$

$\log_e 2$

Floater

+3·8010900, -3
-2.3265280, -2
+6.6953220, -2
-1·2572467, -1
+1·8496307, -1
-2·4645197, -1
+3·3283674, -1
-4·9996399, -1
+9·9999898, -1

Coefficients in power series

Float exponent

Set exponent = 1

Sum power series in y
($y = 2a-1$ where $A = a.2^p$)

$A' = \log(y+1) + (p-1)\log_e 2$

September 1960.

Quicky 15: $A' = \text{Arctan } y/x$

Let $s(z) = +1, \ z \geq 0$
$\quad\quad\quad -1, \ z < 0$

007	33+		Bt = s(y/x)
067	35+		
400	34		A' = s(y) |x|
090	2*		
520	2		
410	34		
430	32		L36 = s(y){|x| - |y|}
410	36		
400	32		L32 = s(y){|x| + |y|}
420	34		
410	32		
300	1		H34 = 1 − p{s(y)[|x| + |y|]}
230	32		
210	34		
370	767		
290	v4		Jump if p ≥ −253
300	15		Error print if $\|x\| + \|y\| < 2^{-253}$
590	9		
-2		(3	
=2,	=852	(12	π
=126,	=402		
=0,	=852	(13	π/4
=126,	=402		
=-8,	=788		+2·3981390, -3
=336,	=314	(14	
=-6,	=0		-1·4152348, -2
=262,	=-464		
=-4,	=253		+3·9345413, -2
=325,	=322		
=-3,	=657		-7·1943845, -2
=325,	=-295		
=-3,	=866		+1·0477539, -1
=163,	=429		
=-2,	=207		-1·4154806, -1
=112,	=-290		
=-2,	=419		+1·9984885, -1
=297,	=409		
=-1,	=157		-3·3332524, -1
=691,	=-342		
=0,	=955		+9·9999987, -1
=1023,	=511	(15	

Coefficients in power series

September 1960.

Quicky 15: (continued)

```
300  749  (4
230  33+
490  2*
330  475
210  35+
300  -1
510  34
430  v3   (2
500  34
410  34
510  32
380  v2
450  2
500  34
420  34
510  36
410  36
500  36
410  34
400  v14
300  #2v14n15
500  34
427  v15
380  -2*
500  36
420  v13
090  3*
520  2
420  v12
200  33+
290  2*
420  v12
```

$A' = 1/\{s(y)[|x| + |y|]\}$

$L36 = \dfrac{|x| - |y|}{|x| + |y|}$

$L34 = \left\{\dfrac{|x| - |y|}{|x| + |y|}\right\}^2$

Sum power series for arctan z $(-1 < z < 1)$

Add $\frac{\pi}{4}$

Correct for quadrant

September 1960.

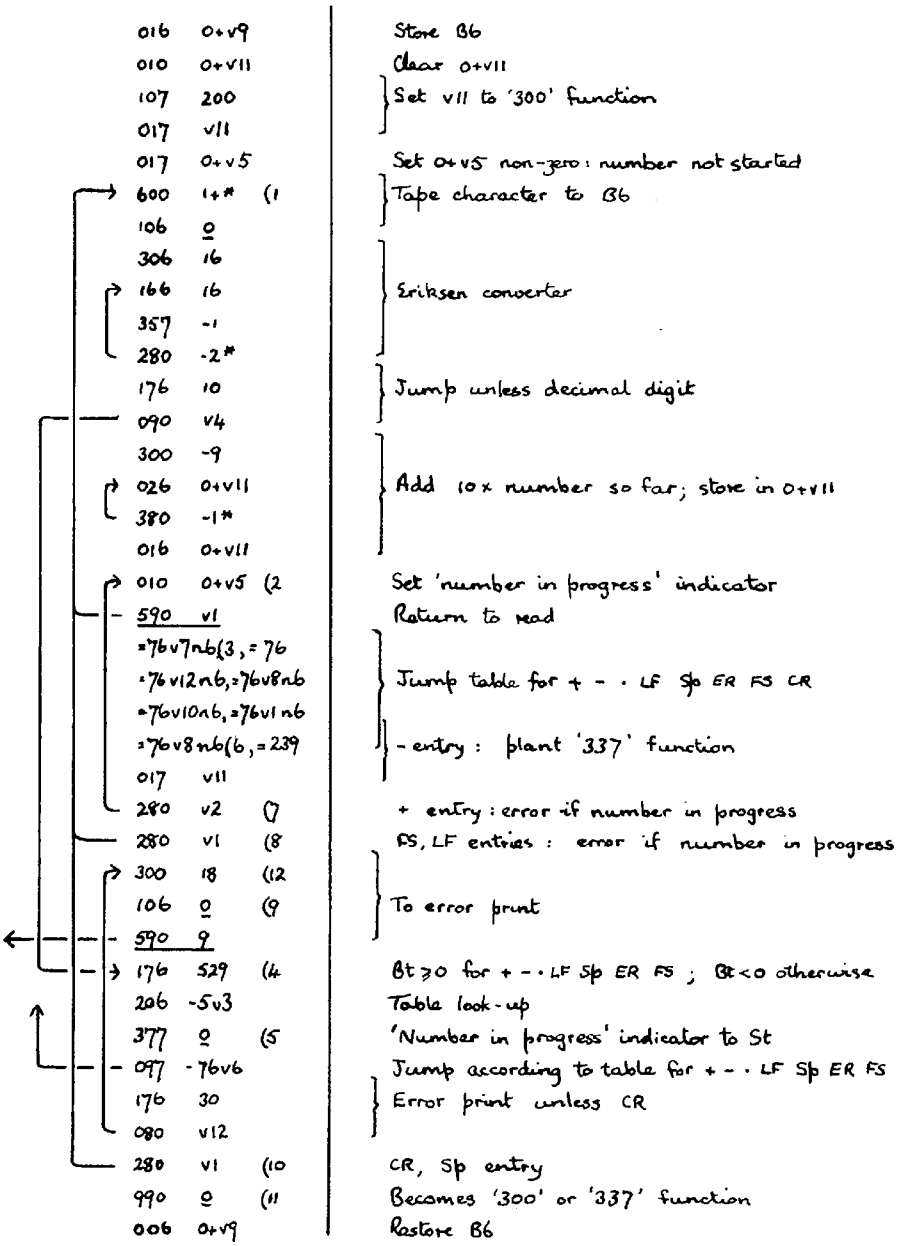
Quicky 16 : A' = Arcsin A

	300 16		S = 16
	490 v1		
	377 0		A' = |A| ; St ≠ 0 if A ≥ 0, St = 0 if A < 0
	520 2		
→	700 511	(1	Jump if p + 511 ≥ 0 i.e. p < 1
	090 v8		
	520 2	(5	
	770 0		Jump with A' = -|A| if |A| = 1
←	090 v10		
←	590 9		To error print if |A| > 1
	=-5, =983	(2	+2·8594730, -2
	=507, =468		
	=-6, =395		+1·4186990, -2
	=900, =464		
	=-5, =533		+2·9249580, -2
	=230, =479		
	=-4, =433		+4·7683409, -2
	=637, =390		
	=-3, =1008		+1·0610532, -1
	=621, =434		
	=0, =100	(3	+6·3661975, -1
	=972, =325		
	=1, =852		π/2
	=126, =402		
→	420 34	(7	
	420 2		
	137 1		
→	700 0	(8	A' = x' = 2x² - 1 with count in B7 until $-\frac{1}{2} < x \leq \frac{1}{2}$
	410 32		
	500 32		
	410 34		
←	090 v7		
	017 0+v4		
	107 -4		
	400 v2		
→	500 34	(9	Power series for $\frac{2}{\pi}\sin^{-1}x$
	427 v3		
←	187 v9		
	510 32		
	450 2		$\frac{2}{\pi}\,2^{-n}\cos^{-1}A$
	707 -16		
	127 0	(4	Adjust exponent with B7
	717 0		
	440 2		$\pm\frac{2}{\pi}\sin^{-1}A$
→	510 2v3	(10	$\pm\sin^{-1}A$
→	280 3v10		Adjust sign with St
→	520 2		

Coefficients in power series

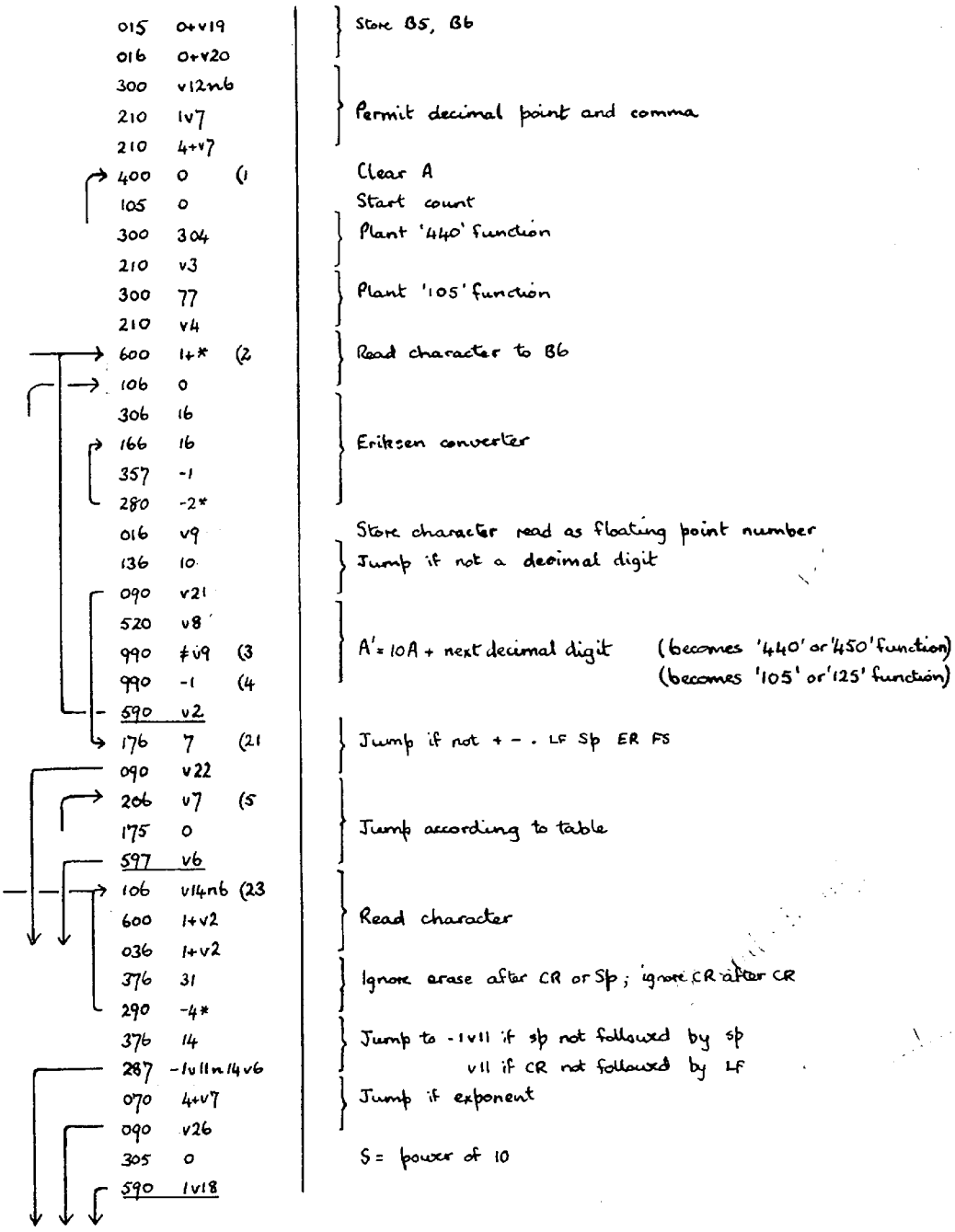
September 1960.

Quicky 18: Read Integer to Sac



September 1960.

Quicky 19: Read fixed or floating point decimal number to the accumulator.



September 1960.

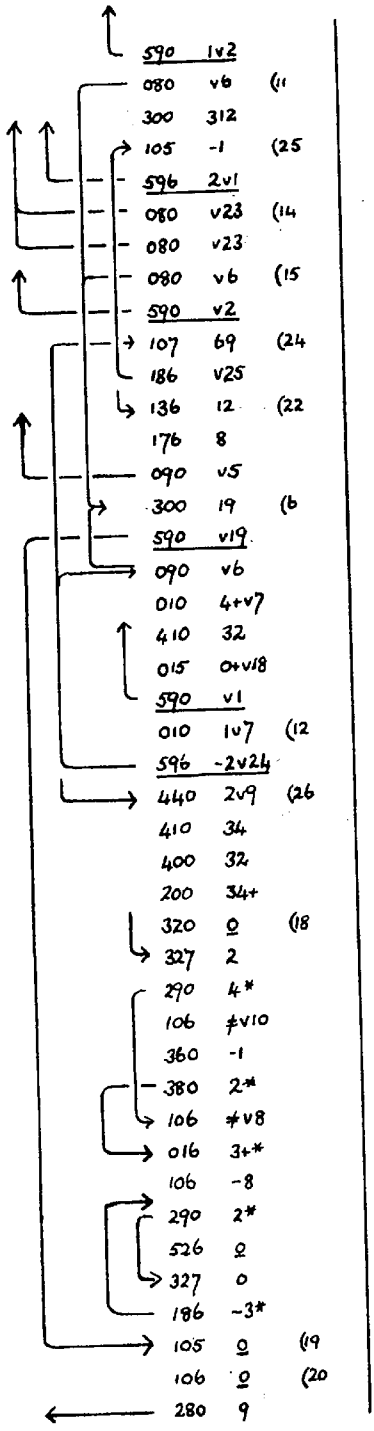
Quicky 19 : (continued)

```
                =213, =1001
                =960, =388
                =107, =437
                =557, =315
                =54,  =312
                =222, =284
                =27,  =0
                =481, =381
                =14,  =0
                =512, =312
                =7,   =0
                =0,   =400
                =4(8, =0
                =0,   =320
                =9,   =0
                =0,   =0      (9
                =29,  =0
                =0,   =384
                =-212, =325
                =1023, =336
                =-106, =982
                =392, =415
                =-53, =664
                =172, =461
                =-26, =738
                =611, =343
                =-13, =747
                =440, =419
                =-6,  =328
                =696, =327
                =-3(10, =410
                =614, =409
=-13 , =-13     =v11n6(7, =v11n6
=0 , =-7        =0,   =v15n6
=-9 , =-80      =v14n6, =v2n6
=-7, =0         =v15n6, =0
=-2, =0         =v14n6, =0
```

} $+10^{64}$
} $+10^{32}$
} $+10^{16}$
} $+10^{8}$
} $+10^{4}$
} $+10^{2}$
} $+10$
} Floater
} Destandardiser
} $+10^{-64}$
} $+10^{-32}$
} $+10^{-16}$
} $+10^{-8}$
} $+10^{-4}$
} $+10^{-2}$
} $+10^{-1}$
} Jump table for + - . LF Sp ER FS CR ,

September 1960.

Quicky 19 (continued)



Jump for LF, Sp
Σ entry : error if number in progress
S = '450' function
Indicate 'number in progress'
Jump ; plant '450' function for -ve number
Entry for Sp
Entry for CR
FS, LF
Ignore Sp, CR, FS, LF if number not started
Set '125' function
Set B6=3 ; jump

} Jump if CR or comma

} Prepare for error print

Error if number not started
Forbid comma
Plant argument
Store -(n+1) : n = no. of digits after decimal point

Entry for ., ,: forbid decimal point
Jump to v24(.), 7v24(,).

} Destrandardise exponent

A' = argument
S = exponent
Add -(n+1)
S = twice total exponent

} Mark and negate negative exponent

} Mark positive exponent

} Multiply by powers of 10 or 1/10

} Restore B5, B6

To error print

September 1960.

ROUTINE INDEX. -1.

The figure shift table, R 15, is on page 36. The look-up is done in R1.

The letter shift table, R 45, is on page 96. The look-up is done in R 81.

ROUTINE INDEX - 2.

© Ferranti Ltd 1960.

Ferranti Ltd

COMPUTER DEPARTMENT

Enquiries to:

London Computer Centre, 68–71 NEWMAN STREET, LONDON, W.1

Telephone MUSeum 5040

and

21 PORTLAND PLACE, LONDON, W.1

Office, Works, and Research Laboratories:

WEST GORTON, MANCHESTER, 12, Telephone EASt 1301

Research Laboratories:

LILY HILL, BRACKNELL, BERKS.