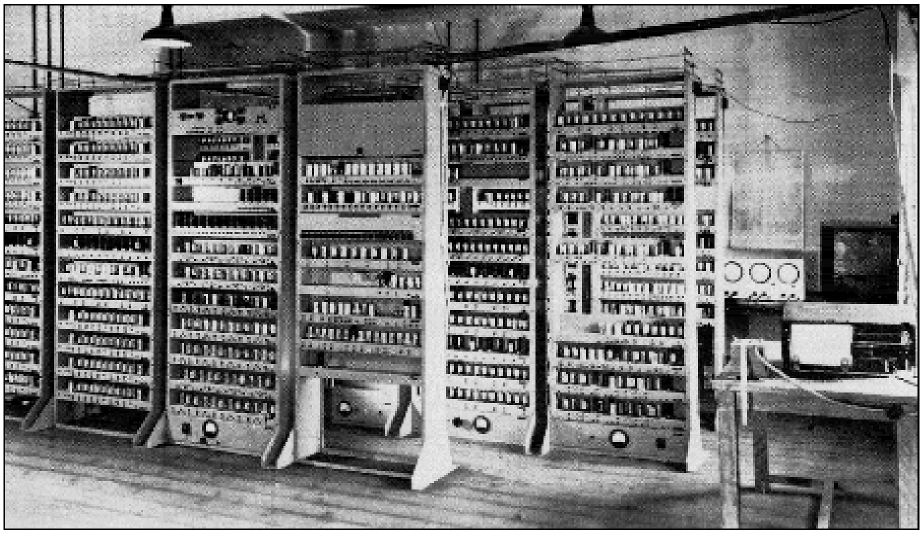
EDSAC PROGRAM DOCUMENTATION



CONTENTS

July 1996

[Corrected December 1996]

PROGRAM NOTATION

The following notation is used on all library program sheets.

Entry points:	If control may arrive at an order by being transferred there by an E or G order the location of the latter (relative to the first order of the subroutine) is shown on the extreme left, with an arrow pointing to the address of the order to which control is transferred, e.g., $16 \rightarrow 23 \quad T\ 6\ \theta.$
Unconditional transfers of control:	A horizontal line is drawn underneath every E or G order which is intended to produce a transfer of control each time it is encountered.
Variable orders:	Orders and pseudo-orders which are to be changed during the course of the calculation are shown in brackets.
Pseudo-orders:	A double vertical line is drawn on the left of the contents of all storage locations which are intended never to be obeyed as orders.
Use of J:	When reading the address part of an order the initial orders treat the letter J as a digit of value 10. Some subroutines therefore use J for the address 10, thus saving one row of holes on the tape.
Preset parameter	C(45), C(46) ... when used as preset parameters are referred to as H parameter, N parameter ...
Control combinations:	Any "order" with code letter K or Z is a control combination. The more common ones are described in ["Control Combinations"].

[Source: WWG 1951, p. 104]

CODE LETTERS FOR INITIAL ORDERS 2

When an order such as A 50 F or A F is being transferred from the tape to the store, the first character to be read is the function letter, and the corresponding binary number is placed by the initial orders in a suitable location for temporary storage. The next character may be either a digit of the address or a code letter F or D. These can be distinguished by the fact that F and D correspond to binary numbers which are greater than ten. The character just read is therefore tested by having 11 subtracted from it; if the result is negative the character must represent a digit of the address, otherwise it represents a code letter. As the successive digits are read the address is built up progressively in binary form. When the code letter is encountered the address and the number representing the function letter are added together. If the code letter is F the result represents the complete order and is transferred to the store as it stands. If the code letter is D, 2^{-16} is added to the result before it is transferred to the store.

In addition to the code letters F and D so far referred to, there are thirteen other code letters which may be used to terminate an order. The object of these code letters is to facilitate the use of subroutines. Each causes the contents of a certain storage location to be added to the order before it is transferred to the store. The complete list of code letters is as follows:

Code letter	Location whose content is added to the order	Number added
F	41	zero
θ	42	variable
D	43	2^{-16}
ϕ	44	
H	45	
N	46	
M	47	
Δ	48	
L	49	variable
X	50	
G	51	
A	52	
B	53	
C	54	
V	55	

Storage location 41 contains zero, so that the code letter F leaves the order unchanged. Storage location 43 contains 2^{-16}, so that code letter D causes 2^{-16} to be added to the order. These two code letters thus have the effect described earlier.

All the above code letters indicate the end of an order, and cause it to be placed in its correct location in the store. The code letter π causes 2^{-16} to be added to the order (in this it resembles D) but must be followed by another code letter to indicate the end of the order. It is thus possible by using π to cause both 2^{-16} and some other number to be added to the order before it is put away in the store.

[Source: WWG 1951, p. 16, with a correction]

SPECIFICATIONS OF LIBRARY SUBROUTINES

Each subroutine is distinguished by a letter denoting its category and a serial number within that category. The categories are as follows.

Category	Subject
A	Floating point arithmetic.
B	Arithmetical operations on complex numbers.
C	Checking.
D	Division.
E	Exponentials.
F	General routines relating to functions.
G	Differential equations.
J	Special functions.
K	Power series.
L	Logarithms.
M	Miscellaneous.
P	Print and layout.
Q	Quadrature.
R	Read (i.e., Input).
S	nth root.
T	Trigonometrical functions.
U	Counting operations.
V	Vectors and matrices.

In the specifications ... the following information is given in abbreviated form immediately beneath the title of each subroutine:

1. Type of subroutine, i.e., whether open, closed, interpretive, or special.
2. Restriction on address of first order. If the word "even" appears it denotes that the first order must have an even address; if no note appears it indicates that the address may be either odd or even.
3. Total number of storage locations occupied by the subroutine.
4. Addresses of any storage locations needed as working space by the subroutine.
5. Approximate operating time (not possible to state in all cases).

[Source: WWG 1951, p. 72]

C7 Check function letters, with localized print suppression.
Special; 61 storage locations; time, see Note 5.

Performs a given program order by order, and prints the function letters of those orders which are drawn from certain specified parts of the store; other orders are obeyed silently. The store may be divided into four regions, orders in two of which have their function letters printed.

Preset parameters:

45	H	P b F		See Note 1
46	N	P (c-a) F		
47	M	P (c-b) F		
48	Δ	P θ	print low	See Note 1.
or		Δ θ	print high	
49	L	P m F	start at m	

Notes: 1. The regions of the store are specified by the parameters a, b, c as follows:

(i) $n < a$
(ii) $a \le n < b$
(iii) $b \le n < c$
(iv) $c \le n$

The subroutine will either "print low," i.e., print function letters of orders in (i) and (iii), or "print high," i.e., print function letters of orders in (ii) and (iv).

2. Print routines in the original program must be arranged to lie in regions from which the function letters are not printed. Characters printed by such routines will appear as figures.

3. A new line of printing is begun at each transfer of control; a clear line is left where orders have been obeyed silently unless such orders themselves cause printing to appear on this line.

4. C7 only tests the locations of orders at each transfer of control, so that if control enters a new region during a consecutive sequence, the mode of operation does not change immediately.

5. Speed of operation is about 5 orders per second when printing function letters, 30 orders per second when suppressed.

6. C7 must be placed at the end of the orders on the tape. After being read it will direct control to itself and commence checking at order m.

	T	Z
0	(Δ	F)
1	(P	F)
2	Q	F
3	A	F
4	θ	F
5	Δ	F
6	π	F
7	K 3000	F
8	P	H
9	P	N
10	P	M

33 →	11	U 26 θ	store address of C.O.	
	12	S 8 θ		
	13	E 15 θ		
	14	A 9 θ	test for change of mode	
13 →	15	S J θ		Transfer
	16	(E 46 Δ)*		control
	17	E 20 Δ		
58 →	18	O 4 θ	new line	
	19	O 5 θ		
17 →	20	U 37 θ	clear top of accumulator	
	21	S 37 θ		
	22	A 3 θ		
45 →	23	A 26 θ		
	24	U 26 θ		
	25	S 26 θ		
	26	(A F)	S.O.	
	27	U 37 θ		
Enter →	28	A θ		
	29	S 3 θ		
	30	(O 37 θ)	becomes E 34 θ for suppression	
	31	E 34 θ		Checking
	32	A 2 θ		cycle,
	33	E 11 θ		similar to
30,31 →	34	U θ		that employed
	35	S θ		in C11
	36	A 1 θ		
	37	(K3000 F)	C.O.	
	38	U 1 θ		
	39	G 41 θ		
	40	A 5 θ		
39 →	41	S 1 θ		
	42	U θ		
	43	S θ		
	44	A 2 F		
	45	E 23 θ		
16 →	46	O 6 θ	figure shift	
	47	E 49 Δ		
16 →	48	O 7 θ	letter shift	
47 →	49	U 37 θ		
	50	S 37 θ		Change of mode of operation
	51	S 16 θ		from printing to suppressed
	52	A 59 θ		or vice versa
	53	U 16 θ		
	54	S 16 θ		
	55	S 30 θ		
	56	A 60 θ		
	57	U 30 θ		
	58	E 18 θ		

```
         G       Z
59       C    35 θ      = C 94 θ θ
60       S    12 θ      = S 71 θ θ
         G       K
         W  2015 Z      = E 28 Z:  stops reading of tape and directs
         E       L        control to order 28 with E L in the accumulator.
```

* Order 16 takes the following forms:

	Printing		Suppressed
Print low	E 46 θ	<----->	G 48 θ
Print high	G 46 θ	<----->	E 48 θ

[Source: WWG 1951, pp. 79-80, 118-20]

C10 Numerical check with delayed start and suppression of check during closed subroutines.

Special; even; 37 + 51 storage locations; time = 1/5 sec per digit printed.

May be applied to a routine in order to print C(Acc) before obeying T orders. It has a delayed start and will cease checking during each closed subroutine. It may be used only on programs containing subroutines with at most one program parameter. If the program has the order A n F in S(n) for a purpose other than entry to a closed subroutine, C10 will fail at that point.

Preset parameters:

45	H	P h F	see Note 1
46	N	P n F	number of digits to be printed
47	M	P m F	address of order at which checking starts.

Notes:

1. Part of the subroutine, 51 orders long, is placed in locations h to (h+50) and may be written over a print routine in the master routine in which case printing from the master routine will be suppressed.
2. A new line of printing is started at each transfer of control.
3. A line feed occurs when a closed subroutine is encountered.
4. The address m of the order at which checking starts must be chosen as described in Note 2 of C7.
5. The first number printed by C10 is the numerical representation of the order at which checking starts.
6. C10 must be placed at the end of the tape and followed by E p K P F, directing control to the master routine.
7. A T order immediately following a closed subroutine with no program parameters will not cause C(Acc) to be printed.

Note: Code letter H refers to locations in the first part of the subroutine and θ to locations in the second part.

		Order	Comment	
		E 25 K		
		T H		
H	0	A 3 F		
	1	T F	dummy print routine	
	2	E F		
10H →	3	O 2 θ	print +	
	4	E 14 H		
31θ →	5	S 6 θ	form A p F/D if order T p F/D is encountered	
	6	E 32 θ		
	7	S 2 H		
	8	T 9 H		
	9	(π F)	becomes A p F/D	Print number transferred by T order
	10	E 3 H	test sign	
	11	T πθ	change sign	
	12	S πθ		
	13	O H	print -	
4H →	14	T πθ		
	15	S 33 H	set digit count in 9 H	
30H →	16	A 2 F		
	17	T 9 H		
	18	A πθ	multiply by 10/16	
	19	R 1 F		
	20	S πθ		
	21	R D		
	22	A πθ		
	23	U θ		

```
         24   O      θ    print
         25   F      θ
         26   S      θ
         27   L    4 F
         28   T     πθ
         29   A    9 H    digit count
         30   G   16 H
         31   T     πθ    clear accumulator
         32   E   34 θ    to sequence control

         33 ‖ P       N    number of digits
160θ → 34   A    2 H    test for order A n F in
         35   S   12 θ     S(n), i.e., S.O. = C.O.
         36   G   19 θ
         37   S    2 F
         38   E   19 θ                                      Test for
         39   O    3 θ    line feed                          entry to
         40   A   20 θ                                      closed
         41   A   12 θ    form A n+2 F                       subroutines
         42   S   26 θ                                      and obey
         43   U   12 θ                                      them
         44   U   47 H                                      directly
         45   S    5 θ    form G n+1 F
         46   T   50 H
         47  (P      F)    = C(Acc.) or A n+2 when
         48   T   22 θ    subroutine is encountered
         49   A   40 H
         50  (P      F)    sign of C(Acc.) or G n+1 F when
                           subroutine is encountered
```

When an order A n F is encountered in n, the order in (n+2) is placed in the C.O. position and control is transferred to (n+1) with A 20 θ in the accumulator. Since there is a G order in (n+1) control is transferred to the subroutine and the link which is planted in the subroutine is E 22 θ (or E 23 θ if the subroutine has one program parameter). When the operation of the subroutine is finished control is transferred to order 22θ (or 23θ) of C10 and checking recommenced.

```
              T       Z
θ          0 ‖ (P      F)    working space
           1 ‖ (P      F)     for print cycle
           2 ‖ Z       F
           3 ‖ Δ       F
           4 ‖ θ       F
           5 ‖ Q     1 F
           6 ‖ Q       F
           7   A       M    extracts order at which checking
           8   T    47 H     starts and replaces it by order
           9   A    15 θ     directing control to C10 (order 21θ)
          10   T       M
          11   O     4 θ    carriage return
          12   O     3 θ    line feed
          13   O     9 H    figure shift
          14   E    25 F
```

```
          15  || E   21 θ
                 E    7 Z
                 P      F
```

The orders 7 to 14 are executed once during input, and then written over by:

```
                 T    7 Z
  18θ  →   7     O    4 θ      carriage return
           8     O    3 θ      line feed
           9     S    2 H      form A n F when control is transferred
  36θ  →  10     U   12 θ
          11     S   12 θ
          12    (G2047 M)      = A -1 M, becomes select order (S.O.)
          13     U   22 θ
          14     A   50 H
          15     S    2 H                                                Checking
          16     G   34 H      test for transfer of control              cycle
          17     S    6 θ                                                similar
          18     G    7 θ                                                to that
36H,38H→  19     U   50 H                                                employed
          20     S   50 H                                                in C11.
Enter  →  21     A   47 H      Add "C(Acc.)"
          22    (T      M)     current order (C.O.)
          23     U   47 H      transfer "C(Acc.)"
          24     E   26 θ
          25     S    3 θ      test C(Acc.) for sign,
  24θ  →  26     S   47 H       if - send 1/2 to 50H
          27     U   50 H
          28     S   50 H
          29     A   22 θ      examine C.O. and test
          30     S    1 H       for T order
          31     E    5 H
   6H  →  32     U   22 θ
          33     S   22 θ
  32H  →  34     A   12 θ
          35     A    2 F      sequence control
          36     G   10 θ
```

During the course of this subroutine the 17 most significant digits of C(Acc.) are stored in 47 H and are restored when an order from the original program is executed.

[Source: WWG 1951, pp. 81, 120-2]

D6 Division, accurate, fast.

Closed; 36 storage locations; working positions 6D and 8D; time = (10m+120) msecs, where $2^{-m-1} \le |C(4D)| \le 2^{-m}$.

Forms C(0D)/C(4D) where C(4D) ≠ 0 and ≠ -1, and places result in 0D.

Accuracy: maximum error is $\pm K \cdot 2^{-35} \pm 2^{-34}$, where K = quotient.

$a_{n+1} = a_n - c_{n+1}a_n + c_{n+1}$

$c_{n+1} = -a_n b + (b-1)$, where b is the shifted divisor

$i - a_n \to 1/b$

$c_n \to 0$ a_n and c_n are negative

$a_0 = 2b - 2\sqrt{2} + 1$; therefore c_n is negative until process is completed

		G		K	
	0	A	3	F	plant link
	1	T	34	θ	
7 →	2	S	4	D	make divisor positive and change sign of quotient
	3	E	13	θ	
	4	T	4	D	
	5	S		D	
	6	T		D	
	7	E	2	θ	
14 →	8	T	4	D	shift divisor and dividend until divisor exceeds capacity
	9	A		D	
	10	L		D	
	11	T		D	
	12	A	4	D	
3 →	13	L		D	
	14	E	8	θ	
	15	R		D	
	16	U	4	D	b-1 to 4D
	17	L		D	
	18	A	35	θ	
	19	T	6	D	a_0 to 6D
	20	E	25	θ	
30 →	21	U	8	D	c_{n+1} to 8D
	22	N	8	D	$-c_{n+1}.a_n$
	23	A	6	D	$+a_n$
	24	T	6	D	$+a_{n+1}$ to 6D
20 →	25	H	6	D	a_{n+1} to multiplier register
	26	S	6	D	a_n
	27	N	4	D	$-(b-1)\cdot a_n$
	28	A	4	D	$+(b-1)$
	29	Y		F	
	30	G	21	θ	test
					<—— accumulator contains 2^{-34}
	31	S		D	form quotient
	32	V		D	
	33	T		D	
	34	(E		F)	link
	35	‖ W1526		D	$3 - 2\sqrt{2}$

[Source: WWG 1951, pp. 83, 125-6]

E2 Exponential, slow.

Closed; 19 storage locations; working positions 0D and 6D; time = 930 msecs.

Forms exp [C(4D)] - 1 and places the result in 4D, $-1 \leq C(4D) < 0{\cdot}693$.

Accuracy: probable error = 2^{-33}.

(e^{x-1}) to 4D, where x = C(4D)

Uses a recurrence relation $z_{n-1} = z_n + z_n^2/2^{n+1}$ starting with $z_{33} = x$ and ending with $z_0 = (e^x-1)$

		G	K	
	0	A	3 F	plant link
	1	T	18 θ	
	2	Y	F	2^{-n} to 6D
	3	L	D	
16 →	4	T	6 D	
	5	H	4 D	form z_n^2
	6	V	4 D	
	7	T	D	
	8	H	6 D	
	9	V	D	$z_n^2/2^n$
	10	R	D	$z_n^2/2^{n+1}$
	11	A	4 D	$z_n + z_n^2/2^{n+1}$
	12	Y	F	
	13	T	4 D	z_{n-1} to 4D
	14	A	6 D	shift strobe
	15	L	D	
	16	E	4 θ	test strobe for end of cycle
	17	T	D	
	18	(E	F)	link

[Source: WWG 1951, pp. 83, 126]

M3 Print heading.

Closed; 10 storage locations (temporarily); working position 0.

Copies information directly from the tape to the teleprinter and may thus be used to print a heading at the top of a sheet.

Notes: 1. M3 is placed at the front of the program tape unless R9 is used, in which case M3 follows R9. No control combinations need precede M3.

2. M3 is immediately followed by the heading, which may include line feed, carriage return, etc., according to the teleprinter code.

3. The heading is followed by blank tape, and the succeeding orders should be prefaced by a control combination of the form P K T n K.

```
              | P     F |
              | G     K |
Enter, 6, 8 → 0 | I     F |   input next symbol
            1 | A     F |  ┐
            2 | R     D |  │ shift ready for printing
            3 | L     F |  │
            4 | U     F |  ┘
            5 | G     F |    print
            6 | E     θ |  ┐
            7 | A   6 F |  │ test for blank tape
            8 | G     θ |  ┘
            9 | E   8 F |    return to initial input
              | E     Z |
              | P     F |
```

[Source: WWG 1951, p. 91; Cambridge University Archives COMP B, 3 June 1950]

M20 Set parameter value, by means of telephone dial, during input of orders.

Special; uses no storage space.

If M20 is included at the appropriate point on the input tape, the H-parameter may be set to $d \cdot 2^{-15}$ by dialing an integer d. As soon as the first few rows of M20 have been read the machine stops on a Z-order. Exactly three decimal digits should then be dialed to specify d.

This subroutine consists largely of control combinations. It requires no storage space, but uses 22F, 42F, and 43F, normally occupied by orders of the initial input routine, as working space.

Notes: 1. A preset parameter other than H may be set by changing the control combination T 45 K near the end of the M20 tape.

2. If it is desired to dial more, or less, than three digits the central section of M20 should be repeated an appropriate number of times, or omitted, as the case may be.

Tape Entry

```
P       Z
Z       K   ] Stop machine; when digit r is dialed set Transfer
M 2037  F   ]   Order to T(r - 10)F
G       K     Copy address (r - 10) into 42
P    10 K     Set C(22) = P 10 F
P       Z     Add C(22) to 42 if (r - 10) < 0; if (r - 10) = 0
                leave unaltered
T    43 K   ] Transfer C(42) to 43
P       θ   ]
Z       K   ]                                        ]
M 2037  F   ]                                        ]
G       K   ] Repeat for second digit dialed         ]
P    10 K   ]                                        ] Central
P       Z   ]                                        ]   section
T    43 K   ] Multiply C(43) by 10, add C(42),       ]
P π   0 θ   ]   and place sum in 43                  ]
Z       K   ]
M 2037  F   ]
G       K   ] Repeat for final digit dialed
G       K   ]
P    10 K   ]
P       Z   ]
T    45 K   ] Multiply C(43) by 10, add C(42), and place sum in 45
P π   0 θ   ]
I    43 K   ] Reset C(43) to P D, Transfer Order to T 46 F, and
B     2 F   ]   resume normal action of initial input routine
      Q     ]
```

[Source: WWG 2nd ed., 1957, pp. 154, 190-1]

P1 Print a single positive number (without layout or round-off).

Closed; 21 storage locations; time = (171n+10) msecs.

Prints the positive number in 0D to n places of decimals, leaving R·10^n in 0D, where R is the remainder.

Program parameter:

p	A p F	orders calling in P1
p+1	G s F	
p+2	P n F	

Notes: 1. Teleprinter must be on figure shift.
2. Layout must be separately controlled.
3. Round-off is not included.

		G K	
	0	A 18 θ	Plant link
	1	U 17 θ	
	2	S 20 θ	Plant S m+2 F
	3	T 5 θ	
	4	H 19 θ	
	5	(P F)	(1) -n x 2^{-15} to Acc. (2) Count digits.
16 →	6	T 5 θ	Digit cycle
	7	V D	Multiply
	8	U F	Print
	9	O F	
	10	F F	Check and remove
	11	S F	
	12	L 4 F	Shift
	13	T D	
	14	A 5 θ	Count digits
	15	A 2 F	
	16	G 6 θ	
	17	(E F)	Link
	18	U 3 F	
	19	J F	= 10/16
	20	M 1 F	

[Source: WWG 1951, p. 92; Cambridge University Archives COMP B]

P6 Print short positive integer.

Closed; 32 storage locations; working positions 1, 4, and 5; time = about 900 msecs.

Prints $2^{-16} \cdot C(0)$ with suppression of nonsignificant zeros but without layout.

		G	K		
	0	A	3 F	Plant link	
	1	T	25 θ		
	2	H	29 θ		
	3	V	F	Multiply by $2^{16}/10^5$	
	4	T	4 D		
	5	A	3 θ	V F = -1/16 to S(0)	
	6	T	F		
	7	H	30 θ	Set multiplier	
	8	S	6 θ	Set digit count	
24 →	9	T	1 F	Digit count	Digit cycle
	10	V	4 D	Multiply	
	11	U	4 D		
	12	A	F	Test for first	
	13	G	26 θ	non-zero digit	
	14	T	F	Clear Acc. and S(0)*	
	15	T	F		
	16	O	5 F	Print	
	17	A	4 D		
	18	F	4 F	Check and remove	
	19	S	4 F		
28 →	20	L	4 F	Shift	
	21	T	4 D		
	22	A	1 F		
	23	S	3 θ	Count digits	
	24	G	9 θ		
	25	(E	F)	Link	
13 →	26	S	F	Add 1/16	Suppress zero
	27	O	31 θ	Space	
	28	E	20 θ		
	29	J	995 F	$\cong 2^{16}/10^5$	
	30	J	F	= 10/16	
	31	φ	F	Space	

* S(0) becomes cleared when the first non-zero digit is encountered, thus preventing the suppression of later zeros.

[Source: WWG 1951, pp. 92; Cambridge University Archives COMP B]

P7 <u>Print positive integer up to 10 digits</u>.

Closed; even; 35 storage locations; working position 4D; time = approx. 1.8 sec.

Prints $2^{34} \cdot C(0D)$ with zero suppression but without layout.

<u>Notes</u>: 1. Teleprinter must be on figure shift.
2. Layout must be separately controlled.
3. C(0D) must be positive and less than $10^{10} \cdot 2^{34}$.
4. If the number to be printed is less than 10^9, the left-hand zeros are replaced by spaces. In any case, 10 positions on the paper are used.

		G	K		
	0	A	3 F	plant link	
	1	T	26 θ		
	2	H	28πθ		
	3	N	D		
	4	Y	F	multiply by $2^{34}/10^{10}$ and add 2^{-34}	
	5	L	D		
	6	T	4 D		
	7	S	27 θ	-1/32 to 0	
	8	T	F		
	9	H	8 θ	set multiplier	
	10	S	8 θ	set digit count	
25 →	11	T	1 F	digit count	
	12	V	4 D	multiply	
	13	A	F	test for first	
	14	G	31 θ	nonzero digit*	
	15	S	F		
	16	L	D	shift	
	17	U	F	print	digit cycle
	18	O	F		
	19	F	F	check and remove	
	20	S	F		
	21	L	4 F	shift	
34 →	22	T	4 D		
	23	A	1 F		
	24	A	27 θ	count digits	
	25	G	11 θ		
	26	(	)	link	
		T	28πZ		
		P	F	**	
		T	27 Z		
	27	P	1024 F		
	28	P	610 D	= $-2^{33}/10^{10}$	
	29	θ	524 D		
	30	φ	F		
14 →	31	O	30 θ	space	
	32	S	F	add 1/32	suppress zero
	33	L	8 F	shift	
	34	E	22 θ		

* C(0) = -1/32 until first nonzero digit is printed, when C(0) becomes positive, thus preventing the suppression of later zeros.

** These symbols appear on the tape and serve merely to clear 28D, thus ensuring that the sandwich digit between 28 and 29 is zero, before further orders are read.

[<u>Source</u>: WWG 1951, pp. 92, 141-2]

P14 Print signed decimal with round-off and digit check. Layout controlled by program.

Closed; 46 storage locations.

Prints the decimal number in C(0D), rounded-off. Digit spacing, number of digits printed and layout are determined by a program parameter.

```
Preset parameter:    45  | H | A m D         round-off order

                     p       | A p F  ]      orders calling in P14
                     p+1     | G s F  ]
Program parameter: .... p+2  | P x F
                     or      | K 4096+x F  Layout constant:  see note 2.
```

Notes: 1. Figure shift is called during the input of orders.

2. The number of digits and their spacing is determined by the program parameter, which is calculated as in note 5. Carriage return and line feed will occur before the number is printed if K 4096 F is added to this layout constant. Each number is followed by a space.

3. If the F order shows an error a line feed will occur and the next digit printed may be in error.

4. Negative numbers are preceded by a negative sign, positive numbers by a space.

5. The digit layout is determined by the program parameter P x F, where x may be obtained as follows. Imagine the printed characters, including digits and spaces (only single spaces are permissible) laid out in the form below, starting with the most significant digit at the left-hand end. Then add together the numbers below the spaces, and the number above the last digit; the sum is x.

	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
	24576	12288	6144	3072	1536	768	384	192	96	48	24	12	6	

For example: (i) to print 10 digit numbers with spaces after the 3rd, 6th, and 9th digits, x = 6144 + 384 + 24 + 4 = 6556; (ii) to print 8 digit numbers with spaces after the 4th and 5th digits, x = 3072 + 768 + 32 = 3872.

Continued ...

```
                  O   40 K    figure shift during input
                  π      F
                  T      Z

           0      A   45 θ
           1      U    4 θ    form A n+2 F
           2      A   22 θ
           3      T   39 θ    form link
           4     (A     F)    = A n+2 F or layout count
           5      E    8 θ
           6      O   40 θ    carriage return
           7      O   41 θ    line feed
     5 →   8      T    4 θ    layout count in 4θ
           9      A      D
          10      E   15 θ    test sign of C(0D)
          11      T      D    reverse sign
          12      S      D
          13      O      θ    print -
          14      E   16 θ
    10 →  15      O   42 θ    print space
    14 →  16      P      H    round-off order
          17      T      D
          18      H   44 θ
          19      A    4 θ
 35,38 →  20      T    4 θ
          21      V      D    multiply by 10/16
          22      U    1 F
          23      O    1 F
          24      F      F
          25      S      F    print digit and check
          26      G   29 θ
          27      S   43 θ
          28      G   30 θ
    26 →  29      O   41 θ
    28 →  30      A   43 θ
          31      L    4 F
          32      T      D
          33      A    4 θ
          34      L      D
          35      E   20 θ    layout count
          36      O   42 θ
          37      L      D
          38      G   20 θ
          39     (E     F)    link

          40      θ      F    carriage return
          41      Δ      F    line feed
          42      ϕ      F    space
          43      Q      F
          44      J      F
          45      P    2 F
```

[Source: WWG 1951, pp. 94, 143-4]

R1 Input a sequence of signed long decimal fractions.

Closed; 55 storage locations; working positions 0, 1, 4, 5, and 6.

Given a sequence of numbers punched as decimals followed by sign, this subroutine places these numbers in pD, (p+2)D, (p+4)D ... and returns control to the master routine when F appears on tape.

```
Preset parameters:   45 | H  positions are used by subroutine
                     46 | N

                      m | A m F   orders calling in R1.
Program parameter:  m+1 | G s F
                    m+2 | T p D
```

Notes: 1. Decimal point is immediately before first digit punched.
2. Any number of digits up to 10 may be punched; more will exceed the capacity of the accumulator.
3. Blank or erased tape is treated as F.

```
              G      K
              T   45 K
       45F    P   32 θ      H parameter
       46F    P   47 θ      N parameter
              T      Z
         0    A    3 N      plant A m+2 F
         1    U    4 θ
         2    A    4 N      plant link
         3    T    9 H
         4   (A     F)      (i) A m+2 F (ii) digit count
   35 →  5    T      H      plant transfer order
         6    T      D      clear 0D and 4D
         7    T    4 D
         8    A    5 N      reset switch
         9    T   13 θ
        10    H    2 N      set multiplier
        11    S    6 N      set digit count**
  24 →  12    T    4 θ      digit count
        13   (I     F)      or T F when switched*
        14    A      F
        15    S    6 N      test symbol for
        16    E    4 H       +, -, or F
        17    T    6 F      clear accumulator
        18    V    4 D      multiply previous digits            digit cycle
        19    L    8 F      shift
        20    A      D      add new digit
        21    T    4 D
  46 →  22    A    4 θ
        23    A    7 N      count digits
        24    G   12 θ
        25    H    4 D
        26    N      N
        27    R  128 F
        28    R  128 F      multiply by 2^34 /10^10
        29    V    1 N
        30    L      D
        31    Y      F
     H  32   (T     F)      transfer to store
     1  33    A      H      change transfer order
     2  34    A    3 N
     3  35    E    5 θ
```

```
16 →  4  36    S    6 N    test for -
      5  37    E   42 θ
      6  38    A    7 N
      7  39    E   44 θ    test for +                  +, - and F
      8  40    T    6 F    F: clear accumulator
      9  41   (E      F)   link

   37 →  42    S    4 D    negative:
         43    T    4 D      change sign               + and -
   39 →  44    A    2 N    set switch to TF
         45    T   13 θ
         46    E   22 θ

      N  47    P  610 D
      1  48    Z 1523 D
      2  49    T      F    =10/32
      3  50    P    2 F
      4  51    U    1 F
      5  52    I      F
      6  53    P    5 D
      7  54    P      D
```

* Order 13 is I F during input of punched digits, T F for dummy zeros which make up remainder of 10 digits.

** Digit count is actually set to 11 because + or - sign is counted as a digit.

[Source: WWG 1951, pp. 96, 146-8]

R2 <u>Input of positive integer during input of orders</u>

Special; 15 storage locations (temporarily);

Reads the input tape and converts the decimal integers thereon to binary form multiplied by 2^{-34} and places these in sequence in storage locations mD, (m+2)D, (m+4)D, etc.

<u>Parameter</u>: T m D must follow the subroutine.

<u>Notes</u>: 1. After the subroutine T m D is punched, followed by the integers, each terminated by F with the exception of the last one which is terminated by π T Z.

2. After the integers have been read, π T Z returns control to the initial orders and subsequent orders read from the tape will be written over R2.

		G K		
9 →	0	T 20 F		
	1	V D	10·(partial sum)*	
	2	L 8 F		
	3	A 40 D	add new digit	
14 →	4	U D	New partial sum to 0D** and to	
	5	(T F)	final destination of number	digit cycle
	6	I 40 F	read next symbol	
	7	A 40 F		
	8	S 39 F	subtract $11 \cdot 2^{-16}$	
	9	G θ	test for F	
	10	S 2 F	subtract $2 \cdot 2^{-16}$	number cycle
	11	G 23 F	test for π (if π return to initial orders)	
	12	A 5 θ	change destination of integer	
Enter →	13	T 5 θ		
	14	E 4 θ		

Followed on tape by:

E 13 Z	on subroutine tape
T m D	punched by user

Hence control enters subroutine at order No. 13, with T m D in the accumulator.

* The multiplier register contains 10/32 throughout input of orders and operation of this subroutine.

** When obeyed for the first time in each number cycle, this order clears 0D.

[<u>Source</u>: WWG 1951, pp. 96-7, 148]

R3 Input of one signed long decimal fraction.

Closed; even; 41 storage locations; working positions 4D and 6D.

Reads one fraction punched in decimal form followed by sign, and places it in 0D.

		G	K		
		T	45 K		
	H	P	26 θ		
		T	Z		
	0	A	3 F	Plant link	
	1	T	H		
	2	T	D	Clear 0D and 4D	
	3	T	4 D		
	4	A	6 H	Reset switch	
	5	T	9 θ		
	6	H	1 H	Set multiplier	
	7	S	4 H	Set digit count**	
80 →	8	T	6 F	Digit count	Digit cycle
	9	(I	F)	or TF when switched*	
	10	A	F		
	11	S	4 H	Test for	
	12	E	7 H	sign symbol	
	13	T	7 F	Clear acc.	
	14	V	4 D	Mult previous digits	
	15	L	8 F	Shift	
	16	A	D	Add new digit	
	17	T	4 D		
40 →	18	A	6 F		
	19	A	5 H	Count digits	
	20	G	8 θ		
	21	H	2πH		
	22	N	4 D	Multiply by	
	23	L	D	$2^{34}/10^{10}$	
	24	Y	F		
	25	T	D	Transfer to 0D	
	H 26	(	)	Link	
		T	28πZ		
		P	F	***	
		T	27 Z		
	1 27	T	F	= 10/32	
	2 28	P	610 D	$= -2^{33}/10^{10}$	
	3 29	θ	524 D		
	4 30	P	5 D		
	5 31	P	D		
	6 32	I	F		
12 →	7 33	S	4 H	Test +	Sign symbol
	34	G	37 θ	or -	
	35	S	4 D	- change sign	
	36	T	4 D		
34 →	37	T	7 F	Clear Acc.	
	38	A	1 H	Set switch	
	39	T	9 θ	to TF	
	40	E	18 θ		

Continued ...

*   Order 9 is IF during input of punched digits, TF for dummy zeros which make up remainder of 10 digits.

**   Digit count is actually set to 11 because sign symbol is counted as a digit.

*** These symbols appear on the tape and serve merely to clear 28D, thus ensuring that the sandwich digit between 28 and 29 is zero, before further orders are read.

[Source: WWG 1951, p. 97; Cambridge University Archives COMP B]

R4 Input of one signed integer.

Closed; 22 storage locations; working positions 4, 5, and 6.

Reads one integer y punched in decimal form followed by sign, and places $y \cdot 2^{-34}$ in 0D.

Notes: 1. $|y| < 2^{-34}$

2. R4 is applicable to either long or short numbers; in the latter case $y \cdot 2^{-16}$ will be left in 0 provided that $-2^{16} \leq y < 2^{16}$.

```
              G       K
          0   A     3 F      Plant link
          1   T    21 θ
          2   T     4 D      Clear 4D
          3   H     6 θ      Set multiplier
          4   E    11 θ
             ------------
          5   P     5 D
          6   J       F      = 10/16
   15 →   7   T     6 F      Clear acc.
          8   V       D      Mult. previous digits
          9   L     4 F      Shift
         10   A     4 D      Add new digit
    4 →  11   T       D      Transfer to 0D              Digit cycle
         12   I     4 F      Read next symbol
         13   A     4 F
         14   S     5 θ      Test for sign symbol
         15   G     7 θ
         16   S     5 θ      Test + or -
         17   G    20 θ                      Sign symbol
         18   S       D      - change sign
         19   T       D
   17 →  20   T     6 F      Clear acc.
         21   (E      F)     Link
             ------------
```

[Source: WWG 1951, p. 97; Cambridge University Archives COMP B]

R9 Input of positive integers during input of orders. Standard form for regular use.

Special: 15 storage locations.

The actual orders of this subroutine are identical with those of R2, but R9 is intended always to be placed in locations 56 to 70 inclusive, and to remain there throughout the input of a whole program, being used any number of times. Each time it is used it will read a sequence of positive decimal integers and place them in consecutive long storage locations.

Notes: 1. The subroutine tape commences with P K T 56 K, so that it may be copied immediately at the head of the tape. It does not have E 13 Z at the end, so that it is not automatically obeyed after being read.

2. R9 is called in by the control combination E 69 K T m D. This is followed by the integers each terminated by F except the last, which is terminated by π to return control to the initial orders. After this must be punched a control combination to restore the transfer order, e.g., T Z. The integers will be placed in mD, (m+2)D, (m+4)D, etc.

3. Negative integers may be read if 2^{35} is added to each before punching.

[Source: WWG 1951, p. 98]

S2 Square root, fast.

Closed; 22 storage locations; working position 0D; time = approx. (36n+180) msecs, where $(2\ 1/4)^{-n-1} \leq C(4D) < (2\ 1/4)^{-n}$.

Forms $\sqrt{C(4D)}$ where C(4D) > 0 and places result in 4D.

Accuracy: Number of significant figures in result is two less than number of significant figures in argument.

Note: 1. If C(4D) = 0, subroutine continues to cycle indefinitely.

Repetitive process:

$a_{n+1} = a_n - 0.5a_nc_n$	$a_0 = C(4D)$	$a_n \to \sqrt{C(4D)}$
$c_{n+1} = c_n^2(0.25c_n - 0.75)$	$c_0 = C(4D) - 1$	$c_n \to 0$

		G	K		
	0	A	3 F	plant link	
	1	T	20 θ		
	2	A	4 D	form c_0	
	3	S	9 θ		
	4	A	6 θ		
19 →	5	U	D	c_n to R	repetitive cycle
	6	H	D		
	7	R	1 F	$(0.25c_n - 0.75)$ to 0D	
	8	S	21 θ		
	9	T	D		
	10	N	4 D	a_{n+1} to 4D	
	11	R	D		
	12	A	4 D		
	13	Y	F		
	14	T	4 D		
	15	V	D	form c_{n+1}	
	16	T	D		
	17	V	D		
	18	Y	F		
	19	G	5 θ	test for $c_{n+1} = 0$	
	20	(E	F)	link	
	21	‖ S	F	= 3/4	

[Source: WWG 1951, pp. 98, 149-50]

S3 Cube root.

Closed; 25 storage locations; working positions 4, 5, 8, and 9; time = approx. 1 sec.

Forms cube root of C(6D) and places result in 0D. C(6D) may be positive or negative and is left unchanged at the end.

Root is formed digit by digit, using a shifting (negative) strobe.

		G	K	
	0	A	3 F	plant link
	1	T	20 θ	
	2	T	D	set first trial (i.e., zero)
	3	S	24 θ	set strobe
19 →	4	T	4 D	
	5	H	D	
	6	V	D	
	7	Y	F	form $(\text{trial})^3$ - C(6D)
	8	T	8 D	
	9	V	8 D	
	10	S	6 D	
	11	E	21 θ	
	12	T	8 D	
	13	S	4 D	increase trial
23 →	14	A	D	
	15	T	D	
	16	A	4 D	
	17	R	D	shift strobe
	18	Y	F	
	19	G	4 θ	
	20	(E	F)	link
11 →	21	T	8 D	
	22	A	4 D	decrease trial
	23	G	14 θ	
	24	‖ I	F	= 1/2

[Source: WWG 1951, pp. 99, 150]

T1 Cosine, rapid.

Closed; even; 44 storage locations; working position 0D; time = 82 msecs.

Forms 0.5 cos[2·C(4D)] where |2·C(4D)| ≤ π/2, and places result in 4D.

Accuracy: maximum error = 2^{-33}.

0 to 14 temporarily	[Subroutine R2]	R2 is included in the T1 tape
	T 32πθ	
32D	1,614 F	coefficients in power series
34D	73,454 F	
36D	2,423,967 F	
38D	54,539,267 F	
40D	763,549,741 F	
42D	5,726,623,061 π	
	T Z	
0	A 3 F	Plant link
1	T 30 θ	
2	H 4 D	Square argument
3	V 4 D	
4	Y F	
5	T 4 D	
6	H 4 D	
7	N 32πθ	
8	A 34πθ	$C(A) = a_{12} - a_{14}x^2$
9	T D	
10	N D	
11	A 36πθ	$C(A) = a_{10} - a_{12}x^2 + a_{14}x^4$
12	T D	
13	N D	
14	A 38πθ	$C(A) = a_8 - a_{10}x^2 + \dots$
15	T D	
16	N D	
17	A 40πθ	$C(A) = a_6 - a_8x^2 + \dots$
18	T D	
19	N D	
20	A 42πθ	$C(A) = a_4 - a_6x^2 + \dots$
21	T D	
22	N D	
23	Y F	$C(A) = -a_4x^2 + a_6x^4 - \dots$
24	T D	
25	N D	
26	S 4 D	
27	A 31 θ	$C(A) = 1/2 - x^2 + a_4x^4 - \dots$
28	Y F	
29	T 4 D	
30	(E F)	Link
31	I F	= 1/2
	T 44 Z	

[Source: WWG 1951, p. 99; Cambridge University Archives COMP B]

ARITHMETIC

This program illustrates various arithmetic instructions on the Edsac simulator.

		T	64	K	Set load point	
	64	Z		F	Stop	
	65	A	96	F	acc = 33	
	66	A	97	F	acc = acc + 46 = 79	Short integer arithmetic
	67	S	98	F	acc = acc - 96 = -17	
	68	T		F		
	69	H	100	F	acc = $^{3}/_{16}$ x $^{7}/_{8}$ = $^{21}/_{128}$	Short fractions
	70	V	101	F		
	71	T		F		
	72	H	104	D	acc = $^{1}/_{3}$ x $^{1}/_{3}$ = $^{1}/_{9}$	
	73	V	104	D		Long fractions
	74	Y		F	Round acc to 34 binary places	
	75	A	106	D	acc = acc - $^{1}/_{9}$ = 0 to 34 b.p.	
	76	T		F		
	77	H	99	F	acc = (5 x 2^{-16})^{2} = 25 x 2^{-32}	
	78	V	99	F		Integer multiplication
	79	L	64	F	acc = acc x 2^{-16} = 25 x 2^{-16}	
	80	L	64	F		
82 →	81	L		D	Left shift till acc -ve	
	82	E	81	F		
87 →	83	R		D		
	84	R		D		Shift operations
	85	R		D	Pretty pattern	
	86	S	103	F		
	87	G	83	F		
		T	96	K	Set load point	
	96	P	16	D	= 33	
	97	P	23	F	= 46	Integer constants
	98	P	48	F	= 96	
	99	P	2	D	= 5	
	100	E		F	0.0011_2 = $^{3}/_{16}$	
	101	K		F	0.1110_2 = $^{7}/_{8}$	Short fractions
	102	Δ		F	1.1000_2 = -$^{1}/_{2}$	
	103	I		F	0.1000_2 = $^{1}/_{2}$	
	104	H	682	D	0.0101... = $^{1}/_{3}$	
	105	T	682	D		Long fractions
	106	K	455	F	0.111000... = -$^{1}/_{9}$	
	107	C	455	F		
		E	64	K	Enter at location 64	
		P		F		

[Author: M. Campbell-Kelly, 1990]

CUBES

Prints a table of the cubes of Nichomachus

Note: Cubes are generated until arithmetic overflow occurs and negative values are produced. The program stops when P6 fails due to trying to print a negative number.

Table of routines

Routine	Location of first order	Number of storage locations occupied
P6 (print)	56	32
Master	88	-

Make-up of program tape

space P K

T 56 K

P6

space P Z

Master

E Z P F

Master routine

		G		K	Set θ-parameter
Enter→	0	Z		F	Stop
	1	O	29	θ	Figure shift
22 →	2	O	30	θ	New line
	3	O	31	θ	
	4	A	23	θ	k to 0F
	5	T		F	
	6	A	6	θ	Print 0F using P6
	7	G	56	F	
P6 →	8	T	23	θ	Zero to k
	9	A	24	θ	
	10	A	27	θ	n+1 to n
	11	T	24	θ	
	12	S	24	θ	-n to count
21 →	13	T	26	θ	
	14	A	25	θ	
	15	A	28	θ	m+2 to m
	16	U	25	θ	
	17	A	23	θ	k+m to k
	18	T	23	θ	

19	A 26 θ	Increment count
20	A 27 θ	
21	G 13 θ	Jump to 13 if count ² 0
22	E 2 θ	Repeat main cycle
23	P D	k (n^3; =1 initially)
24	P D	n (=1 initially)
25	P D	m (=1 initially)
26	P F	count
27	P D	=1
28	P 1 F	=2
29	π F	figs
30	θ F	cr
31	Δ F	lf

[Author: M. Campbell-Kelly, 1990]

RECIPROCALS
Prints the reciprocals of the integers 1 to 10.

Table of routines

Routine	Location of first order	Number of storage locations occupied
D1 (divide)	56	36
P1 (print)	92	21
Master	113	-

Make-up of program tape

space P K

T 56 K

M3

θΔ*RECIPROCALSθΔπ   Table heading

space P Z

T 56 K

D6

space P Z

P1

space P Z

Master

E Z P F

Continued …

Master routine

		G		K	
		T	47	K	Set M parameter
		P	21	θ	
		T		Z	
	0	S	1	M	Set count to -9
19 →	1	T	6	M	
	2	A	2	M	$1 \cdot 2^{-4}$ to 0D
	3	T		D	
	4	A	7	M	$n \cdot 2^{-4}$ to 0F
	5	T	4	D	
	6	A	6	θ	Set 0D to 0D/4D (ie. 1/n)
	7	G	56	F	using subroutine D6
D6 →	8	O	3	M	Output new line
	9	O	4	M	
	10	O	5	M	Output decimal point
	11	A	11	θ	Print 0D
	12	G	92	F	using subroutine P1
	13	P	10	F	Parameter for P1 (10 dec. places)
P1 →	14	A	7	M	Increment n
	15	A	2	M	
	16	T	7	M	
	17	A	6	M	Increment and test counter
	18	A		M	
	19	G	1	θ	
	20	Z		F	Stop
M	0	P		D	= 1
	1	P	4	D	= 9
	2	Q		F	$= 1 \cdot 2^{-4}$
	3	θ		F	carriage return
	4	Δ		F	line feed
	5	M		F	decimal point
	6	P		F	count
	7	W		F	$= n\ (=2 \cdot 2^{-4}$ initially)

[Author: M. Campbell-Kelly, 1990]

HELLO WORLD

Prints "HI" on the teleprinter

```
                T   64 K    Load from location 64
                G      K    Set θ parameter
Start → 0       Z      F    Stop
        1       O    5 θ    Letter shift
        2       O    6 θ    Print "H"
        3       O    7 θ    Print "I"
        4       Z      F    Stop
        5       *      F    Letters
        6       H      F    "H"
        7       I      F    "I"
                E      Z
                P      F    Enter at location 0θ
```

[Author: M. Campbell-Kelly, 1990]

PRINT PRIMES

Prints the primes of the odd integers from 5 up to 4 decimals digits, until stopped by the operator

n = number being tested
m = number being tested as a factor
p = position on line of printed page
d = digit counter

```
         31 | T 107 S | As required by initial input
         32 | O  92 S | Figures
  87 --> 33 | O  93 S | Line feed        New line
         34 | O  94 S | Carriage return
         35 | S   5 S | Set position count, p = -5
         36 | T   6 S |
  86 --> 37 | O  95 S | Double space
         38 | O  95 S |
 106 --> 39 | T   7 S |
         40 | A  96 S |
         41 | R   4 S |
  43 --> 42 | S  97 S | Test whether m is
         43 | E  42 S |  a factor of n (see note 2)
         44 | L   4 S |
  46 --> 45 | A  97 S |
         46 | G  45 S |
         47 | S  98 S |
         48 | G 100 S |
         49 | T   7 S |
         50 | A  97 S | m+2 to m
         51 | A   4 S |
         52 | T  97 S |
         53 | H  97 S |
         54 | N  97 S |
         55 | L  64 S | If m > square root of n
         56 | L  64 S |  then stop testing
         57 | A  96 S |
         58 | E  39 S |
         59 | T   7 S |
         60 | A  96 S | n is prime; transfer to 1S for printing
         61 | U   1 S |
         62 | A   4 S | n+2 to n
         63 | T  96 S |
         64 | A  99 S | 3 to m
         65 | T  97 S |
         66 | S  88 S | Set digit count, d = -4
  83 --> 67 | T   7 S |
         68 | H  91 S |
         69 | A   1 S |
         70 | E  72 S |
  73 --> 71 | V  91 S |
  72 --> 72 | S  89 S |
         73 | E  71 S |
         74 | A  89 S | Print digit
         75 | T     L |
         76 | O     S |
         77 | H  90 S |
         78 | V   1 S |
         79 | L   4 S |
         80 | T     L |
```

	81	A 7 S			
	82	A 98 S	d + 1 to d		
	83	G 67 S			
	84	A 6 S			
	85	A 4 S	p + 1 to p		
	86	G 36 S			
	87	E 33 S			
	88	P 2 S	= 4 (digit count)		
	89	P 500 S	= 1000	Used for binary	
	90	J S	= $^{10}/_{16}$	to decimal	
	91	P 16 S	= 32	conversion	
	92	π S	figure shift		
	93	θ S	carriage return		Constants
	94	Δ S	line feed		
	95	φ S	space		
	96	P 2 L	n (= 5 initially)		
	97	P 1 L	m (= 3 initially)		
	98	P L	= 1		
	99	P 1 L	= 3		
48 →	100	T 7 S			
	101	A 99 S	3 to m		
	102	T 97 S			
	103	A 4 S		If n not a prime	
	104	A 96 S	n+2 to n		
	105	T 96 S			
	106	E 39 S			

Notes

1. The odd numbers, n, beginning from 5 are tested.

2. Testing is done by effecting division by repeated subtraction.

3. Factors tested are 3, 5, 7, ... m, where m need not exceed the square root of n.

4. L or S is treated as the least significant digit.

[5. Location: 4S contains 2; 5S contains 10; 6S contains p; 7S contains d.]

[6. The annotation has been augmented to correspond to the original flow diagram.]

[7. Order 85 has been changed from A 98 S to A 4 S, to correspond to the original specification in which 5 numbers per row are printed.]

[Author: D. J. Wheeler, c.May 1949]

[Source: "The EDSAC Demonstration", Report of a Conference on High-Speed Calculating Machines (1949), reprinted in B. Randell, Origins of Digital Computers 1982, Springer, New York, pp. 423-9. With additional annotation.]

PRINT SQUARES

Prints the squares and first differences of the integers 1 to 100

	31	T 123 S	As required by initial input
enter →	32	E 84 S	Jump to 84
	33	P S	Used to keep count of subtractions
	34	P S	Power of 10 being subtracted
	35	P10000 S	For use in the decimal binary conversion
	36	P 1000 S	
	37	P 100 S	
	38	P 10 S	
	39	P 1 S	
	40	Q S	
	41	π S	Figures
	42	A 40 S	
	43	ϕ S	Space
	44	Δ S	Line feed
	45	θ S	Carriage return
	46	O 43 S	
	47	O 33 S	
	48	P S	Becomes number to be printed
94 →	49	A 46 S	Put O 43 S in 65S
	50	T 65 S	
72 →	51	T 129 S	Clear 129S
	52	(A 35 S)	Put power of 10 in 34S
	53	T 34 S	
	54	E 61 S	Jump to 61
63 →	55	T 48 S	
	56	A 47 S	
	57	T 65 S	To control printing
	58	A 33 S	
	59	A 40 S	
	60	T 33 S	
54 →	61	A 48 S	
	62	S 34 S	Print contents of 48S
	63	E 55 S	
	64	A 34 S	
	65	P S	
	66	T 48 S	
	67	T 33 S	
	68	A 52 S	
	69	A 4 S	
	70	U 52 S	
	71	S 42 S	
	72	G 51 S	
	73	A 117 S	
	74	T 52 S	
	75	(P S)	End print [link]

	76	P S	Becomes x
	77	P S	Becomes x^2
	78	P S	Becomes x^2
	79	P S	Becomes Δx^2
	80	E 110 S	
	81	E 118 S	
	82	P 100 S	
	83	E 95 S	
32 →	84	O 41 S	Set on print figures
120 →	85	T 129 S	Clear 129S
	86	O 44 S	
	87	O 45 S	
	88	A 76 S	x+1 to 76S and 48S
	89	A 4 S	
	90	U 76 S	
	91	T 48 S	
	92	A 83 S	Set switch Z
	93	T 75 S	
	94	E 49 S	
1: 75 →	95	O 43 S	Double space
	96	O 43 S	
	97	H 76 S	$x^2 \cdot 2^{15}$ to 77S
	98	V 76 S	
	99	L 64 S	
	100	L 32 S	
	101	U 77 S	
	102	S 78 S	Δx^2 to 79S
	103	T 79 S	
	104	A 77 S	x^2 to 48S and print
	105	U 78 S	
	106	T 48 S	
	107	A 80 S	
	108	T 5 S	
	109	E 49 S	
2: 75→	110	O 43 S	Double space
	111	O 43 S	
	112	A 79 S	Δx^2 to 48S and print
	113	T 48 S	
	114	A 81 S	
	115	T 75 S	
	116	E 49 S	
	117	A 35 S	
3: 75→	118	A 76 S	Test for finish:
	119	S 82 S	
	120	G 85 S	
	121	O 41 S	
	122	Z S	

[Author: M. V. Wilkes, c.May 1949]

[Source: "The EDSAC Demonstration", Report of a Conference on High-Speed Calculating Machines (1949), reprinted in B. Randell, Origins of Digital Computers 1982, Springer, New York, pp. 423-9. With additional annotation.]

THE TPK ALGORITHM

INTRODUCTION

In their report "The Early Development of Programming Languages" (1976) Knuth and Trabb Pardo argue that the best way to understand a programming language is to study specimen programs; this communicates the flavor of a language far more effectively and concisely than a lengthy programming manual. In their report the authors introduce the TPK algorithm.

The TPK algorithm is a short program that demonstrates many of the characteristic features of a program; by coding TPK in a variety of languages Trabb Pardo and Knuth have been able to contrast a number of historic programming languages in a most succint yet informative fashion.

A version of the TPK algorithm written in Pascal is given below, together with specimen input and output. The TPK algorithm demonstrates the following points: the use of variables, constants and a vector; a program loop proceeding by positive increments and another by negative increments; accessing successive vector elements; a conditional statement; built-in functions, such as square-root and absolute value; input-output procedures; a user written procedure. The program is quite short and would probably have taken between a few seconds and a couple of minutes to run on a first-generation computer, depending on how fast the computer was and how effective the language and its translator.

Of course TPK does not actually do anything useful, but it would be difficult to devise a more illustrative program using fewer statements.

Pascal program:

```
program TPK (input, output);
function f (t: real): real;
  begin
    f := sqrt(abs(t)) + 5 * t * t * t
  end;
var
  i: integer;
  y: real;
  a: array[0..10] of real;
begin
  for i := 0 to 10 do
    read (a[i]);
  for i := 10 downto 0 do
    begin
      write(i : 5);
      y := f(a[i]);
      if y > 400 then writeln (999.0:13:5)
                 else writeln (y:13:5)
    end
 end.
```

Test data:

```
1.5  8  -6  9.5  2.3  9.9
2.1  -2.1  6  0.001  -0.002
```

Printed output:

10	0.04472
9	0.03162
8	999.00000
7	-44.85586
6	47.75413
5	999.00000
4	62.35157
3	999.00000
2	-1077.55054
1	999.00000
0	18.09974

Reference:

D. E. Knuth and L. Trabb Pardo, "The Early Development of Programming Languages", pp. 197-213 of N. Metropolis et al (eds.) A History of Computing in the Twentieth Century, Academic Press, NY, 1980.

THE TPK ALGORITHM FOR EDSAC

Scaling calculation

In the TPK algorithm, for each element t in the vector we have to calculate

$$y = \sqrt{|t|} + 5t^3 \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (1)$$

If we make the assumption that all elements of the vector are less than about 10 in magnitude then we can rewrite (1) as

$$y' = 2^{-11} \cdot \sqrt{|t'|} + 5 \cdot 2^{-1} \cdot t'^3 \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (2)$$

where $y' = 2^{-13}y$ and $t' = 2^{-4}t$. Now all the numbers handled are less than unity.

Table of routines

Sub-routines etc.	Location of first order	Number of storage locations occupied
R1 (read fractions)	56	55
P7 (print integer)	112*	35
P14 (print fraction)	147	46
S2 (square root)	193	22
Auxiliary subroutine	215	23
Master routine	238	-

* first order must be in an even location

Notes:

By convention the first subroutine is placed in location 56 onward, locations 0 to 55 being occupied by the initial orders and the preset parameters. The vector is stored as follows: a_0 in 20D, a_1 in 22D ..., and a_{10} in 40D. (The notation nD means the long location consisting of locations n and n+1.) These locations are in fact occupied by the initial orders during program input, but are overwritten when the program proper assumes control. This was quite a usual practice in order to make the most of the storage.

Master routine

		G		K	[1]	
		T	47	K	Sets M-parameter [2]	Control combinations
		P	38	θ		
		T		Z		
	0	A		θ		
	1	G	56	F		Calls in R1 to read vector a_0, a_1 ... a_{10} into 20D, 22D ... 40D [3]
	2	T	20	D	Parameter	
R1, 35 →	3	O	10	M	Newline	
	4	O	11	M		
	5	T		D		
	6	T		D		
	7	A	7	M	Copies count i into OD and prints it using P7	
	8	T		F		
	9	A	9	θ		
	10	G	112	F		
P7 →	11	O	12	M	Outputs two spaces	
	12	O	12	M		
	13	H	4	M	Scales a[i] by 10/16 [4]	
	14	V	40	D		
	15	T	8	D	Calls in auxiliary subroutine using 8D for argument t' and result y'	
	16	A	16	θ		
	17	G	215	F		
auxil-iary →	18	H	8	D		
	19	A	8	D	Sets multiplier register to y' if y' less than $400\cdot 2^{-13}$, otherwise $999\cdot 2^{-13}$	
	20	S	5	M		
	21	G	23	θ		
	22	H	6	M		
21 →	23	T		D		
	24	V		2πM	[5]	
	25	T		D	Scales multiplier register by $10^{-4}\cdot 2^{13}$, transfers to OD and prints it using P14	
	26	A	26	θ		
	27	G	147	F		
	28	‖ P	3104	F	[6]	
P14 →	29	A	14	θ		
	30	S	9	M	Modify order 14 [7]	
	31	T	14	θ		
	32	A	7	M		
	33	S	8	M	Decrement count and branch to order 3 if positive or zero	
	34	U	7	M		
	35	E	3	θ		
	36	Z		F	Stop	
	37	P		F	Filler, to make next location even	

M	0	P 4 D	$\frac{1}{2}\cdot 10^{-9}$ [8]
	1	P F	
	2	T 1714 F	$10^{-4}\cdot 2^{13}$
	3	Z 219 D	
	4	J F	10/16
	5	P 1600 D	$400\cdot 2^{-13}$
	6	P 3996 F	$999\cdot 2^{-13}$
	7	P 5 F	Count i (+10)
	8	P D	Decrement (+1)
	9	P 2 F	Modifier
	10	θ F	Carriage return
	11	Δ F	Line feed
	12	ϕ F	Space

Notes:

The master routine corresponds to the main program of the Pascal version of the TPK algorithm. Its operation should be reasonably clear from the annotation and the notes below.

1. The top line, G K, is the control combination to set the θ-parameter for relocation.

2. The next three lines are used to set the M-parameter so that all constants used in the program are addressed relative to location m, where m is the value of the M-parameter. The advantage of this is that if the code for the master routine changes in length during the program development process, only the M-parameter has to be changed and the instructions in the program which refer to constants do not have to be altered.

3. Although the original TPK algorithm uses a for-loop to input the vector, there was a vector-read subroutine R1 so we have used it.

4. The subroutine R1 inputs fractions so the data has already been scaled by 10-1; hence the scale factor of 10/16. (The input data is shown below.)

5. The M-parameter is set for short numbers (ie. with the length indicator bit set to zero); the code letter π preceding M overrides this for a long number.

6. The program parameter for P14 controls the print layout; this particular value gives 9 decimal digits with a space between the 4th and 5th positions.

7. Lines 29-31 are particularly interesting: They modify the array-accessing order in line 14 by subtracting 2 from the address so that next time round the loop the array element immediately to the left of the current one is used. This sort of technique had to be used in most early machines until index registers were adopted. Incidentally, the program might be improved slightly if it were made self-initialising; as it is, if it were desired to process another set of data, the program would have to be reloaded to restore the array accessing order to its original state.

8. These two pseudo-order pairs are long constants needed for rounding and for scaling; they were obtained from a list of such useful constants given in Programming Bulletin No. 3 (11 October 1950).

Auxiliary subroutine

```
            G       K
       0    A     3 F    ] Plants link 1
       1    T    22 θ    ]
       2    A     8 D    ]                          ]
       3    E     6 θ    ]                          ]
       4    S     8 D    ] |t'| to 4D 2             ] Calculates √|t'|
       5    S     8 D    ]                          ]
  3 →  6    T     4 D    ]                          ]
       7    A     7 θ    ] Calls in S2 to           ]
       8    G   193 F    ] calculate √4D            ]
            ----------
 S2 →  9    H     8 D    ]
      10    V     8 D    ]
      11    T       D    ]
      12    V       D    ]
      13    R       D    ] Calculates 5·2^-1·t'^3
      14    U       D    ] using add and shift
      15    L     1 F    ] orders 3
      16    A       D    ]
      17    T       D    ]
      18    A     4 D    ]
      19    R   512 F    ] Calculates 2^-11√|t'| + 5·2^-1.t'^3
      20    A       D    ] and stores in 8D
      21    T     8 D    ]
      22    (Z     F)      Return order planted here
            ----------
```

Notes:

The auxiliary subroutine corresponds to the procedure f in the Pascal version of TPK; its job is to evaluate equation (2) above. The subroutine uses 8D for the argument t' and the result y'. Some explanatory notes follow.

1. Lines 0-1 plant the return link for the Wheeler jump in line 22. Note that line 22 is filled with a stop order, Z F; this is so that the program will come to a halt if the return link is put in the wrong place due to a coding error.

2. The coding for absolute value is spelled out in full; the operation was too short to justify inclusion in the subroutine library.

3. Multiplication by powers of two is done by left and right shift orders; this was one of the advantages of scaling in powers of two.

Make-up of program tape

space P K [1]

T 56 K [2] Program goes into location 56 onwards

R1

P F Extra pseudo-order to make first location of P7 even
space P Z [3]

P7

space P Z
G K T 45 K A 276 D [4] Preset parameter for P14

P14

space P Z

S2

space P Z

Aux

Auxiliary subroutine

space P Z

Master

Master routine

space P K [5]

E 238 K P F [6]

Notes:

1. The program tape begins with a length of blank leader tape. The initial orders would make some interpretation of blank tape and the control combination P K overcomes this by resetting the initial orders to the state they were in immediately before the blank tape.

2. The control combination T 56 K causes the following program to be placed in location 56 onwards. (This is broadly equivalent to setting the origin in a modern assembler with a directive such as "ORG 56".)

3. The routines are separated by blank tape so that the individual routines can be identified. Hence the control combination P Z (or P K). (Incidentally, very early on P Z P Z was used, but it was shortly realized that P Z only was sufficient (Programming Bulletin No. 5, 15 January, 1951). The action of the initial orders could be very obscure.

4. The control combinations to set a preset parameter immediately precede a library subroutine. This sets the H-parameter, but we will forgo the details.

5. Some blank tape is left for the insertion of a "jiffy tape" for program corrections at the end of the program.

6. The final control combination causes the program to be entered at location 238. (In a modern assembler we would use something like "ENTER 238".)

NUMBER SEQUENCE FOR INPUT DATA

```
15+8+6-95+23+
99+21+21-6+0001+
0002-F
```

Note:

Numbers are punched as decimal fractions followed by a sign. F is a data terminator.

PRINTED OUTPUT

```
        10   0000 04472
         9   0000 03162
         8   0999 00000
         7  -0044 85586
         6   0047 75414
         5   0999 00000
         4   0062 35157
         3   0999 00000
         2  -1077 55051
         1   0999 00000
             0018 09974
```

Note:

A single space has been left between the fourth and fifth digits of the right-hand column of the tabulation; this is where the decimal point would go when the answers are scaled up by 10^4. Notice also that the zero on the last line of the tabulation is missing; this is due to a programming limitation in the zero-suppression coding of the library subroutine P7.

[Source: M. Campbell-Kelly, "Programming the EDSAC: Early Programming Activity at the University of Cambridge", Annals of the History of Computing, Vol. 2 (1980), pp. 7-36.]

INITIAL ORDERS 1

Notes

1. When the starting button is pressed these orders are placed in locations 0 to 30 and control is set so that the first order obeyed is in location 0.

2. The first order to be punched on the tape must be T n S, where the last order is to be input to position n-1. Control is then automatically transferred to the beginning of the routine after the last order has been input by the initial input routine.

	0	T S	Clears accumulator and puts 10/32 in multiplier register
	1	H 2 S	
	2	T S	Control switched to 6. Locations 0-3 are then used as 'working space'
	3	E 6 S	
	4	P 1 S	2^{-15} — Constants
	5	P 5 S	$10 \cdot 2^{-16}$
3, 30 →	6	T S	Input function digits to their correct digital position in 0
	7	I S	
	8	A S	
	9	R 16 S	
	10	T L	
20 →	11	I 2 S	Reads character on the tape and test whether it is numerically less than 10
	12	A 2 S	
	13	S 5 S	
	14	E 21 S	
	15	T 3 S	Clears accumulator using 3 as a rubbish dump
	16	V 1 S	One stage of the binary-decimal conversion. New partial address is obtained by taking ten times old partial address and adding the next digit
	17	L 8 S	
	18	A 2 S	
	19	T 1 S	
	20	E 11 S	Transfers control to location 11
14 →	21	R 4 S	Control is transferred to 21 from the order 14 when character read is S or L. When L has been read the 17th digit of the accumulator is a 1, when S has been read it is a 0
	22	A 1 S	The address has been formed x 2^{-16} and so needs doubling
	23	L L	
	24	A S	Forms the complete order in the accumulator
	25	(T 31 S)	Transfers the order to its final position in store
	26	A 25 S	Increases the address specified in order 25 by 1; eg. T 31 S is replaced by T 32 S, and so on
	27	A 4 S	
	28	U 25 S	
	29	S 31 S	Tests whether all orders have been taken in. Location 31 contains orders T (n+1) S, the first order ton be placed in the store: and so C(Acc) will be positive only when all orders have been taken into the store
	30	G 6 S	
			End of initial orders
	31	T (n+1)S	The first order to be placed in the store

[Sources: D. J. Wheeler, "Programme Organisation and Initial Orders for the EDSAC", Proceedings of the Royal Society A, Vol. 202, pp.573-89, 1950; and "The EDSAC Demonstration", Report of a Conference on High-Speed Calculating Machines (1949), reprinted in B. Randell, Origins of Digital Computers 1982, Springer, New York, pp. 423-9. With additional annotation.]

INITIAL ORDERS 2

	0	(T F)	These orders cause control to be transferred to 20. They are not used after the start, but their locations are used as working space.
	1	(E 20 F)	
	2	P 1 F	These are constants which are intended to be left here unaltered in any program.
	3	U 2 F	
12 →	4	A 39 F	Input of address. This group of orders is entered at 8 with the accumulator empty, so that 0 is cleared. The next digit on the tape is taken in and tested to see if it is less than eleven; if so it is doubled and added to ten times the content of 0, the sum being sent back to 0. The next digit is read, tested, etc., and this is continued until the whole address has been formed; the next digit read, x, is greater than ten and so corresponds to a code letter.
	5	R 4 F	
	6	V F	
	7	L 8 F	
28, 38 →	8	T F	
	9	I 1 F	
	10	A 1 F	
	11	S 39 F	
	12	G 4 F	
	13	L D	These test to see if x is greater than sixteen. If it is, the order A(24+x)F is formed and planted in 20. If x is sixteen or less a switch order E(16+x)F is formed and planted in 20.
	14	S 39 F	
	15	E 17 F	
	16	S 7 F	
15 →	17	A 35 F	
	18	T 20 F	
	19	A F	This adds the address, which is always positive, into the accumulator.
	20	(H 8 F)	This order places 10/32 in the multiplier register during the start and is later replaced by a manufactured one which either adds to the accumulator the number determined by x, or switches control to an address determined by x.
	21	A 40 F	This adds in the function digits of the order so the accumulator now contains the order from the tape plus the number selected by x.
	22	(T 43 F)	This (the transfer order) transfers the assembled order to its final place in the store.
	23	A 22 F	These orders increase the address specified in the transfer order by unity.
	24	A 2 F	
31 →	25	T 22 F	
	26	E 34 F	Transfers control to 34.
20 →	27	A 43 F	Control is switched to these orders by 20 when π has been read from the tape. They add 2^{-16} to the address (which is in the accumulator) and transfer control to 8. The address now refers to a long storage location.
	28	E 8 F	
20 →	29	A 42 F	This adds the address in 42 to the accumulator.

20 →	30	A 40 F	This adds the function digits of the order to the accumulator. The result is that the number in the accumulator is positive if the order has function digits represented by T or E, while it is negative in the case of G.
	31	E 25 F	If the accumulator is positive, the order in the accumulator replaces the order in 22; if negative the accumulator contains the address specified in order 22 which is then put in 42 (the storage location corresponding to θ).
20 →	32	A 22 F	
	33	T 42 F	
26 →	34	I 40 D	These take in the function digits, shift them to their correct place and transfer them to 40. The order in 35 is also used as a constant.
	35	A 40 D	
	36	R 16 F	
	37	T 40 D	
	38	E 8 F	
	39	P 5 D	A constant used in the input of the address. It equals 11.2^{-16}
	40	(P D)	A constant used during the start. It equals 2^{-16}

When the starting button is pressed, the initial orders are placed in storage locations 0 - 40 and control transferred to). The first orders to be executed are the following:

0	T F	clears accumulator
1	E 20 F	transfers control to 20
20	H 8 F	places 10/32 in multiplier register
21	A 40 F	adds 2^{-16} to accumulator
22	T 43 F	transfers 2^{-16} to 43 (the storage location corresponding to D).
23	A 22 F	increase order 22 to T 44 F
24	A 2 F	
25	T 22 F	

The initial input is now ready to take in orders; the first part of the input tape is blank so that the first code letter is a space which corresponds to 16; control is therefore switched from 20 to 32, and the contents of 22 are transferred to 42. This action will continue, the spaces being treated alternately as function digits and code letters. The first symbols encountered will be P and F. There are two possibilities, either

(1) the last space has been treated as a function digit in which case the word read is "space" Z, which causes the address (n-1) to be placed in 42, where n is the address in the Transfer Order; or
(2) the last space was treated as a code letter, in which case the word read is PZ, which causes the address in 42 to be placed in the Transfer Order.

In either case, the Transfer Order is unaltered and will place the first order read from the tape in 44, unless a control combination to reset the Transfer Order occurs first, as will usually be the case.

[Source: WWG 1951, pp.159-60, with corrections from WWG 1957, pp. 218-20]

Department of Computer Science • University of Warwick • Coventry CV4 7AL • UK