Section 9 Extracodes I200 - I777

This section contains a print-out of the Atlas Extracode programs from I200 upwards in Intermediate Input. These are, apart from residual errors and amendments which may from time to time prove desirable, in the form in which they will be loaded into the London and Harwell Fixed Stores. They are not an exact print-out of what is loaded in the Fixed Store of MUSE. Errors were found and improvement devised in a number of the extracodes in MUSE after they had been loaded, and the necessary changes were made making as few changes to the ''hairbrushes'' as possible. In some cases this involved inserting additional jump instructions. For the London and Harwell machines the instructions have to some extent been re-ordered to reduce the number of jumps and generally to tidy up the routines. At some date it may be possible to change the MUSE Fixed Store to render at least the arithmetic extracode part identical in all machines. However, the basic arithmetic is the same in all the computers.

Sub-section 9.I contains a description of the linking sytem for the functional extracode subroutines. This has been written in ABL. A description giving further information about the methods of the extracodes, particularly the functional ones, will be issued later in the volume containing the routine specifications.

30.8.63

9.I The Interlinking of the Functional Extracodes

The routines for implementing the following extracodes are all interconnected:-

I400	ca' = log s:
I402	ca' = exp s:
I410	ca' = sq.rt. s:
I411	am' = arg s:
I412	am' = mod. s:
I413	ca' = s cos s*, s sin s*
I700	am' = log s
I701	am' = log aq
I702	am' = exp s
I703	am' = exp aq
I710	am' = sq.rt. s
I711	am' = sq.rt. aq
I712	am' = sq.rt. (aq^2+s^2)
I713	am' = am*
I720	am' = arcsin s
I721	am' = arcsin aq
I722	am' = arccos s
I723	am' = arccos aq
I724	am' = arctan s
I725	am' = arctan aq
I726	am' = arctan (aq/s)
I730	am' = sin s
I731	am' = sin aq
I732	am' = cos s
I733	am' = cos aq

These extracodes use five basic subroutines, namely:-

1. Square Root
2. Arctan, arccot
3. Log
4. Exp
5. sin, cos

In the cases of 2 and 5 the required function is indicated by means of markers in various B-lines.

All these subroutines are closed, i.e. exit is by means of a link-setting. Links are carried in B97.

Thus a simple exit is I2I I26 97 0

For a simple extracode which only requires the use of a single subroutine, for example, I720 to I725, this exit will be to a 'dummy exit' instruction 52I 0 0 0. In other cases, however, where operations are required afterwards, the exit will be to other routines. For example, I7I3 ($am^{(1)}$) requires first that log am be formed, then the result multiplied by s, and finally the exponential of this product formed. In such a case, to save instructions to reset links, a system is used whereby a single setting of B97 will normally cause the correct exit through all relevant routines.

The following is an outline of the complete system (in ABL notation). The entry points for all the extracodes are indicated, and all the link-setting and link-implementing instructions are shown. Also, an indication is given of the formulae used for the extracodes which call for more than one subroutine.

These routines are not listed in the order in which they occur in the store; they are listed in an order which seems logical in order to expound the system of interlinking. Since the labelling system used is sequential throughout the extracodes it is easy to discover the absolute position of each group of instructions.

In the annotations, the following notation is sometimes used for convenience, in addition to the standard Atlas notation:

x and y for s and s* (i.e. the real and imaginary parts of s:)

u and v for c(ba) and c(ba+I) (i.e. the real and imaginary parts of the complex accumulator Ca.)

9.I continued

JUMP TABLES

I2I	I26	0	A670	I400	ca' = log s: u = log$\surd(x^2+y^2)$, v = arctan (y/x)
I2I	I26	0	A469	I402	ca' = exp s: u = exp x cos y, v = exp x sin y
I2I	I26	0	A625	I4I0	ca' = sq.rt.s: u = $\surd(\frac{1}{2}(\surd(x^2+y^2)+x))$, v = y/2u
I2I	I26	0	0.IA662	I4II	am' = arg s: am' = arctan (y/x)
I2I	I26	0	0.IA626	I4I2	am' = mod s: am' = $\surd(x^2+y^2)$
I2I	I26	0	0.IA469	I4I3	ca' = s cos s*, s sin s*
324	0	II9	0	I700	am' = log s Set aq' = s
I2I	I26	0	A587	I70I	am' = log aq.
324	0	II9	0	I702	am' = exp s Set aq' = s
I2I	I26	0	A364	I703	am' = exp aq
324	0	II9	0	I7I0	am' = $\surd$s Set aq' = s
I2I	I26	0	0.IA629	I7II	am' = $\surd$aq
I2I	I26	0	0.IA627	I7I2	am' = $\surd(aq^2+s^2)$
I2I	I26	0	A56I	I7I3	am' = am^c am' = exp (s log am)
324	0	II9	0	I720	am' = arcsin s Set aq' = s
I2I	I26	0	A678	I72I	am' = arcsin aq am' = arctan (s/$\surd(I-x^2)$)
324	0	II9	0	I722	am' = arccos s Set aq' = s
I2I	I26	0	0.7A678	I723	am' = arccos aq am' = arccot (s/$\surd(I-x^2)$)
324	0	II9	0	I724	am' = arctan s Set aq' = s
I2I	I26	0	A650	I725	am' = arctan aq
I2I	I26	0	0.IA663	I726	am' = arctan (aq/s)
324	0	II9	0	I730	am' = sin s Set aq' = s
I2I	I26	0	0.IA544	I73I	am' = sin aq
324	0	II9	0	I732	am' = cos s Set aq' = s
I2I	I26	0	A544	I733	am' = cos aq

ROUTINES

625)	I2I	97	0	-3A640	(I4I0)	Set link to exit to A640
	- - - - - - - - - - -					
626)	- - - - - - - - - - -				(I4I2)	Set s* in A to form (s^2+s^{*2}) in A
	- - - - - - - - - - -					
627)	- - - - - - - - - - -				(I7I2)	Form (a^2+s^2) in A
	- - - - - - - - - - -					
629)	2I0	97	I26	-3A669	(I7I0/I)	Set link to exit to A669 if not already set
630)	- - - - - - - - - - -					
63I)	- - - - - - - - - - -					SQUARE ROOT
	- - - - - - - - - - -					Form $\surd$a in A
	I2I	I26	97	3		Exit to b97+3

9.I continued

640)	- - - - - - - - - - - -				(I4IO continued)
	- - - - - - - - - - - -				Add x, multiply by I/2
	- - - - - - - - - - - -				
	I2I	97	0	-3A643	Set link
	I2I	I26	0	A63I	Jump to form $\sqrt{(\frac{1}{2}(\sqrt{(x^2+y^2)}+x))}$, exit to A643
643)	- - - - - - - - - - - -				
	- - - - - - - - - - - -				Store as u, form y/2u, store as v
	756	I22	95	0	EXIT from I4IO
650)	- - - - - - - - - - - -				(I724/5)
	I2I	96	0	0	Set marker for arcTAN
	I2I	97	0	A669	Set link to exit to A669
652)	- - - - - - - - - - - -				
	- - - - - - - - - - - -				ARCTAN/COT
	- - - - - - - - - - - -				Form arctan/cot a in A
	I2I	I26	97	0	EXIT
652)	334	0	II9	I	(I4II) Set s* in a to form s*/s
663)	2IO	97	I26	A669	(I726) Set link to exit to A669 if not already set
	- - - - - - - - - - - -				
	- - - - - - - - - - - -				Form am/s
	I2I	I26	0	A652	Jump to form arctan
669)	52I	0	0	0	EXIT
670)	I2I	97	0	A68I	(I400) Set link
	I2I	I26	0	A626	Jump to form $\sqrt{(x^2+y^2)}$, exit to 3A68I = A683
672)	235	I26	0	A676	(I720, I, 2, 3 continued) Jump if $\sqrt{(I-x^2)} \neq 0$
	- - - - - - - - - - - -				If = I, jump direct to arctan
	I2I	I26	0	A652	exit to A669
676)	- - - - - - - - - - - -				
	- - - - - - - - - - - -				(I720, I, 2, 3 continued) if $\sqrt{(I-x^2)} \neq 0$,
	- - - - - - - - - - - -				form $x/\sqrt{(I-x^2)}$
	I2I	I26	0	A652	Then jump to form arctan. Exit to A669

9.I continued

678)	- - - - - - - - - - - -				(I720, I, 2, 3)
	- - - - - - - - - - - -				Form $I-x^2$
	I2I	97	I26	A669-(*+I)	Set b97 = A669, preserving marker from bI26
	I65	96	97	0.2	Set marker for arcsin/arccos
	236	I26	0	A630	Jump to form $\sqrt{(I-x^2)}$ if $I-x^2 \geq 0$. Exit to A669=A672
	Error exit if $I-x^2 < 0$ i.e. $x^2 > I$				
681)	356	I22	0	I	(I300 continued) Store arctan (y/x) as v
	334	0	0	4J7	Recover $\sqrt{(x^2+y^2)}$
	I2I	I26	0	A589	Jump to form $\log\sqrt{(x^2+y^2)}$. Exit to A4I4
683)	356	0	0	4J7	(I400 continued) Store $\sqrt{(x^2+y^2)}$
	I2I	I26	0	A662	Jump to form arctan y/x. Exit to A68I
4I4)	756	I22	0	0	(End of I400) Store $\log\sqrt{(x^2+y^2)}$ as u. EXIT
587)	I2I	97	0	A537-A4I4+A68I	(I700/I) Set link to exit to A537
588)	- - - - - - - - - - - -				Standardize
589)	- - - - - - - - - - - -				LOG
	- - - - - - - - - - - -				Form log a in A
	I2I	I26	97	A4I4-A68I	EXIT
537)	52I	0	0	0	EXIT
56I)	I2I	97	0	A563-A4I4+A68I	(I7I3) Set link
	235	I26	0	A588	Jump to form log am if a ≠ 0, exit to A563
	- - - - - - - - - - - -				
	- - - - - - - - - - - -				If a = 0, prepare to set am' = 0 or E0
	- - - - - - - - - - - -				depending on whether s ≥ or <0
	I2I	I26	0	A575	Jump to complete, exit to A537
563)	342	0	II9	0	(I7I3 continued) Form s x log am
564)	I2I	97	0	A563-A4I4+A68I	(I702, 3 join) Set link to exit to A537
565)	- - - - - - - - - - - -				
	- - - - - - - - - - - -				EXP
	- - - - - - - - - - - -				Form exp a in A
575)	- - - - - - - - - - - -				
	- - - - - - - - - - - -				(tail) set E0 if out of range
	I2I	I26	97	A537-A563+A4I4-A68I	Exit

9.I continued

```
469)  324   0     II9   0          (I402, I4I3)  Set x in Am
      I2I   97    0     A47I-A537+A563-A4I4+A68I   Set link
      2II   I26   I26   A565       Jump if I402 to form exp x.  Exit to A47I
47I)  356   0     0     2J7        (Here directly with x in Am if I4I3; exp x if I402)Sto1

      324   0     II9   I          Set s* in Am
      I2I   I26   0     A545       Jump to form cos s*.  Exit to A584

544)  I2I   97    I26   -(*+I)-A584+A47I+A563-A4I4+A68I
                                   Set link to exit to A537.  Preserve marker in b126
545)  - - - - - - - - - - - - -    SIN/COS
      - - - - - - - - - - - - -    Form sin a in A if b97 odd, cos a if even
      I2I   I26   97    A584-A47I+A537-A563+A4I4-A68I   Exit

584)  362   0     0     2J7        (I402, I3 continued)  Multiply by x (I4I3) or
                                                         exp x (I402)
      356   I22   0     0          Store as u
      324   0     II9   I          Bring out y
      I2I   97    0     2.I*-A584+A47I-A537+A563-A4I4+A68I   Set link to exit to 2*
      I2I   I26   0     A545       Jump to form sin y, exit to I*
      362   0     0     2J7        Multiply by x (I4I3) or exp x (I402)
      756   I22   0     I          Store as v and exit
```

```
(0)=5I2*4
    2II,  I26,  I24,  (I)      |I200 ba'=n if AO set, clear AO. Jump when Acc. free
    2II,  I26,  I24,  2(I)     |I20I ba'=n if AO not set, clear AO. Ditto
          +0    /     0
 96)      *4    /     0        |Floating-point zero
    I2I,  I26,  0     (4)      |I204 ba'=no. of identical chars. from m.s. end of g and s
          +0    /     0
    I65,  9I,   98,   *77      |I206 ba'=n if m.s. char. of g=0. Extract m.s. char.
    2I5,  I2I,  I,    0        |)Set ba'=n if = 0, otherwise 'set BO' Exit
    52I,  I22,  II9,  0,       |)
I6) I02,  II9,  I27,  -0.4     |(I216) Subtract N From BII9, i.e. bII9'=bm
    2I7,  I26,  II9,  (97)     |)Jump to exit if bm ≤ 0
    2I4,  I26,  II9,  (97)     |)
I7) 50I,  I22,  I27,  -0.4     |Otherwise set ba=N and exit
          +0    /     0
    I2I,  I26,  0,    (I6)     |I2I6 ba'=n if bm > 0
    I02,  II9,  I27,  -0.4     |I2I7 ba'=n if bm ≤ 0. bII9'=bm
    2I7,  I26,  II9,  (I7)     |)Jump to set ba=N and exit if bm≤ 0
    2I4,  I26,  II9,  (I7)     |)
    52I,  0,    0,    0        |Otherwise exit
    I0I,  9I,   0,    6*6      |I223 ba'=n if Bcarry = I. Extract V6
    2II,  I2I,  9I,   0        |)Set ba'=n if l.s. digit = I. Exit
    52I,  I22,  II9,  0
    I24,  I26,  0,    0.5      |I226 ba'=n if bt>0. Add 0.5 to BI26
    227,  I26,  I26,  2.3      |I227 ba'=n if bt ≤ 0. Jump 3 or 4 if bt < 0
    224,  I26,  I26,  I.3      |Jump 2 or 3 if bt=0
    2II,  I2I,  I26,  0        |Set bI2I=0 if I227
    52I,  I22,  II9,  0        |Set ba'=n if ≤ 0 (I227), >0 (I226). Exit
97) 52I,  0,    0,    0        |(I2I6 and I226 if bt ≤ 0), Exit
    I2I,  I26,  0,    0.I(34)  |I234 c'=c+2 if am approx = s
    I2I,  I26,  0,    (34)     |I235 c'=c+2 if am not approx=s
    I24,  I26,  0,    0.5      |I236 ba'=n if am>0. Add 0.5 to BI26
    237,  I26,  I26,  2.3      |I237 ba'=n if am≤0. Jump 3 or 4 if am < 0
    234,  I26,  I26,  I.3      |Jump 2 or 3 if am=0
    2II,  I2I,  I26,  0        |Set bI2I=0 if I237
    52I,  I22,  II9,  0        |Set ba'=n if ≤ 0(I237) >0(I236). Exit
    52I,  0,    0,    0        |(I236 if am ≤ 0) Exit
77)    *77  /  *0077           |Character masks
       *000077 /  7.7
 54)     +0   /  7.7
    *00007777 /  *00777777
    I2I,  I26,  0,    (50)     |I250 ba'=char. s in bits 0.5
    I2I,  I26,  0,    (5I)     |I25I s'=char. in bits 0.5 of ba
    203,  I26,  II9,  (52)     |I252 Unpack n chars. Jump if n ≠ 0, reduce n by I
    203,  I26,  II9,  0.I(52)  |I253 Pack n chars. Ditto
    52I,  0,    0,    0        |Exit if n = 0
    356,  0,    0,    (99)     |I255 ba' = n if m ≠ 0 or I's. Store m
    357,  0,    0,    I(99)    |Store 1
    I2I,  I24,  0,    *0I4     |Set exponent = I2
    340,  0,    0,    (0)      |Standardize, i.e. shift up 39 or more if m = 0 or all I's
    2I7,  I2I,  I24,  0        |Set bI2I=0 if shifted 39 or more places
    334,  0,    0,    (99)     |Recover m
    344,  0,    0,    I(99)    |Recover 1
    52I,  I22,  II9,  0        |Set ba'=n if shifted 39 or more otherwise 'set BO'Exit
    I25,  98,   0,    0        |I265g'=(64)g+n,ba'= overflow. Shift ms½
    I25,  99,   0,    0        |Shift l.s.½
    I65,  9I,   987.7          |Extract formertopchar.fromB98(overflow)
    I27,  98,   0,    *777777  |Remove it from bottom of B98
    I64,  98,   99,   7.7      |Add former top char. of B99 into bottom of B98
    I27,  99,   0,    *777777  |Remove it from bottom of B99
    I24,  99,   II9,  0        |Add in n to B99
    I0I,  92,   0,    6*6      |)Add in I at bottom of B98 if adding in n
    I64,  98,   92,   0.I      |)                                    set Bcarry
    52I,  I22,  9I,   0        |Exit putting overflow in ba.
          +0    /     0        |Fixed point zero
```

```
1) I2I,  9I,  0,    I.0     |(I200)
   II6,  9I,  0,    6*6     |Reverse A0 setting for I200
   I0I,  9I,  0,    6*6     |(I20Ijoins) Extract V6
   I27,  9I,  0,    I.0     |Mask out A0 digit
   II6,  9I,  0,    6*6     |Clear A0
   2I5,  I2I, 9I,   0       |Set bI2I=0 if A0 digit = 0 (I200),=I(I20I)
   52I,  I22, II9,  0       |Set ba or b0=n. Exit
4) I2I,  92,  0,    -7      |(I204) Set count
   I2I,  9I,  98,   0       |Copy m.s. ½ of g to B9I
   I06,  9I,  II9,  0       |Non-equivalent with m.s. ½ of s
   2I5,  I26, 9I,   4(0)    |Jump if different
   I2I,  9I,  99,   0       |If same, copy l.s. ½ of g to B9I
   I06,  9I,  II9,  0.4     |Non-equivalent with l.s. ½ of s
   I2I,  92,  0,    -3      |Set counter
   I65,  93,  9I,   *77     |Extract m.s. character
   2I5,  I26, 93,   4(0)    |Jump if non-zero
   I25,  9I,  0,    0       |Shift round
   20I,  I26, 92,   -3(0)   |Cycle back reducing counter
   I2I,  92,  0,    I       |Set b92 =I when all 8 chars. same
   52I,  I22, 92,   7       |Exit setting up ba
50) I0I,  9I,  II9,  0       |(I250) Extract s
   2I0,  I26, II9,  2(0)    |Jump if k=I or 3
   I25,  9I,  0,    0       |Shift round if k=0 or 2
   I63,  II9, II9,  0       |bII9'=½s-s
   2II,  I26, II9,  3(0)    |Jump if even, i.e. last 2 digits of s same
   I25,  9I,  0,    0       |)Shift twice more, i.e. 3 in all for k=2
   I25,  9I,  0,    0,      |)                and 2 in all for k=I
   565,  I22, 9I,   7.7     |Extract bottom char. which is required one and exit
51) II3,  I22, 0,    (99)    |(I25I) Store ba
   I0I,  9I,  0,    (99)    |Set into B9I
   I65,  92,  II9,  0.3     |Extract k
   I24,  92,  92,   0       |)Shift up twice and subtract I.4
   I24,  92,  92,   -I.4    |)i.e. b92=0,-0.4,-I.0,-I.4, for k=3,2,I,0
   2I4,  I26, 92,   4(0)    |Jump if = 0, i.e. k=3 (no shifting needed)
   I23,  93,  92,   0.4     |Set b93=-(b92+0.4)=0,0.4,I.0, for k=2,I,0
   I25,  9I,  0,    0       |Shift round ba    ) shift I,2,3
   202,  I26, 93,   -I(0)   |Cycle, counting in b93    ) for k=0,I,2
   I06,  9I,  II9,  0       |Non-equivalent with s
   I07,  9I,  92,   I.4(77) |Mask out required character position
   5I6,  9I,  II9,  0       |Non-equivalent back into s, i.e. plant new char. Exit
52) II3,  I22, 0,    (99)    |(I252,3) Store ba
   I24,  I2I, 0,    0.4     |Step on BI2I to point at Ba*
   I0I,  9I,  0,    (99)    |Set ba in B9I
   I65,  93,  9I,   0.3     |Extract char. position of start,=k
   II3,  I22, 0,    (99)    |Store ba*
   I24,  93,  93,   0       |)
   I24,  93,  93,   -I.4    |)b93'=-I.4,-I.0,-0.4,0 as k=0,I,2,3
   I0I,  92,  0,    (99)    |Set ba* in B92
   I0I,  94,  9I,   0       |Set c(ba) in B94
   I04,  I26, 93,   *40053I24|Modified jump to shift 0,I,2,3 as k=0,I,2,3
   I25,  94,  0,    0
   I25,  94,  0,    0
   I25,  94,  0,    0
   I67,  9I,  0,    0.I     |Force marker at bottom of B9I
   2II,  I26, I26,  (95)    |Jump if I252
   I07,  94,  93,   I.4(54) |(I253 continued) Remove characters to be replaced
   I05,  94,  92,   0       |Shift required char. posn. to bottom add c(ba*)
   I24,  92,  0,    0.4     |Step on ba*
   200,  I26, 93,   4(0)    |Jump if b93 ≠ 0 (not last char. of ½ word), add 0.4.
   II3,  94,  9I,   0       |If end of ½ word, store b94 back in C(ba)
   200,  93,  9I,   -I.4    |     step on ba and reset count,
   I2I,  94,  0,    0       |     and clear B94 for next chars.
   203,  I26, II9,  -5(0)   |Cycle, counting characters
   I23,  95,  93,   I.4     |b95'=-(b93+I.4)
```

30.8.63

```
      2I4,  I26,  95,   (97)      |(I253 continued) Jump to exit if last char. fills ½ word
      I0I,  96,   95,   2(54)     |Extract mask
      II7,  96,   9I,   0         |Clear required char. positions in c(ba)
      I04,  I26,  95,   *40053I3  |Modified jump to shift 3,2 or I as final k=0,I or 2
      I25,  94,   0,    0
      I25,  94,   0,    0
      I25,  94,   0,    0
      5I6,  94,   9I,   0         |Plant into c(ba) and exit
      200,  I26,  93,   3(0)      |(I252) Jump if b93≠0 (not last char. of ½ word), subtr.I
      200,  93,   9I,   -I.4      |if end of ½ word, step on ba and reset counter
      I0I,  94,   9I,   0         |Extract new c(ba)
95)   I25,  94,   0,    0         |(Entry) Shift char.to foot of B94
      I65,  95,   94,   7.7       |Extract character
      II3,  95,   92,   0         |Store in c(ba*)
      I24,  92,   0,    0.4       |Step on ba*
      203,  I26,  II9,  -7(0)     |Cycle, counting characters
      52I,  0,    0,    0         |Exit
            +0    /     0
34)   356,  0,    0,    (99)      |(I234,5) Store am
      234,  I26,  I26,  5.0       |Jump 6 places if am=0
      32I,  0,    II9,  0         |Subtract s
      374,  0,    0,    (99)      |Divide by 'am'
      366,  0,    0,    (0)       |Take modulus
      32I,  I22,  0,    0         |Subtract C(ba)
      237,  I26,  I26,  0.I       |Add 0.I to control if <0
      2I0,  I26,  I26,  2(0)      |Jump if BI26 odd, i.e. approx = for I235, not for I234
      I24,  I27,  0,    I.0       |Add I to BI27 otherwise
      734,  0,    0,    (99)      |Recover am and exit
```

```
     (0)=768*4
5)   334,  0,    II9,  0        |I300 ba'=int.pt.s, am'= frac.pt.s. Put s in A
     I2I,  I26,  0,    (I)      |I30I ba'=int.pt.am, am'=frac.pt.am. Jump
     I20,  II9,  0,    0        |I302 ba'=ba.n.  Set bII9'=-n
     I2I,  I26,  0,    (3)      |I303 ba'=-ba.n
     2I5,  I26,  II9,  0.5(3)   |I304 ba'=int.pt(ba/n), b97'=rem. Jump if n≠0
     376,  0,    0,    (5)      |Cause DO interrupt if n=0
           0     /     0
           0     /     0
           0     /     0
           0     /     0
     I20,  II9,  0,    0        |I3I2 ba'=ba,n (24 bit integers) Set bII9'=-n
     I2I,  I26,  0,    0.I(3)   |I3I3 ba'=-ba.n (ditto)
     2I5,  I26,  II9,  0.4(3)   |I3I4 ba'=int.pt.(ba/n), b97'=rem. (24 bit int.) Jump n≠0
     376,  0,    0,    (5)      |Cause DO interrupt if n=0
I6)  335,  0,    0,    (99)     |(I302,3,I2,I3 continued) Set am'=-n
     352,  0,    0,    I(99)    |Multiply by ba
I8)  2I0,  I26,  95,   2(0)     |(I304,I4 rejoin) Jump if I304,I2,I3
     365,  0,    0,    (0)      |Shift down if I302,3,I4
     357,  0,    0,    (99)     |Store
     2I6,  I26,  II9,  2(0)     |Jump if ba and n are same sign
     II2,  0,    0,    0.4(99)  |Negate answer if opposite
     334,  0,    0,    2(99)    |)Restore A
     344,  0,    0,    3(99)    |)
     50I,  I22,  0,    0.4(99)  |Set result in ba and exit
95)        *06404 /    0        |Mantissa I/I6,exponent 26
I)   356,  0,    0,    (99)     |(I300,I) Store
     2I7,  I24,  I24,  0        |ay'=0 if ay<0
     300,  0,    0,    (95)     |Add number (m=I/I6, e=26),ie. shift integer to bottom of 1,
     357,  0,    0,    I(99)    |Store al=int.pt       |then standardize, i.e. shift up
     3II,  0,    0,    (95)     |Subtract number off   |one octal place, so octal fraction
     302,  0,    0,    (99)     |am'=frac.part.        |clear.
     50I,  I22,  0,    I.4(99)  |ba'=int.part. Exit
     I20,  II9,  0,    0        |I340 Shift ba down n,(arithmetic, unrounded). bII9'=-n
     2I5,  I26,  II9,  (40)     |I34I Shift ba up n (arithmetic). Jump if n≠0
     I20,  II9,  0,    0        |I342 Shift ba down n (circular). bII9'=-n
     2I5,  I26,  II9,  (42)     |I343 Shift ba up n (circular). Jump if n≠0
     I20,  II9,  0,    0        |I344 Shift ba down n (logical). bII9'=-n
     2I5,  I26,  II9,  0.I(40)  |I345 Shift ba up n (logical). Jump if n≠0
     52I,  0,    0,    0        |(I340-5) Exit if n=0
     II3,  I22,  0,    (99)     |I347 h'=h v ba. Store ba
     I0I,  9I,   II9,  0        |h
     I47,  9I,   0,    (99)     |v ba
     5I3,  9I,   II9,  0        |Store as h' and exit
     I2I,  I26,  0,    (53)     |I353 ba' = posn of m.s. I bit of n.
44)  2I6,  I26,  92,   (97)     |(Logical shift down) Jump if n>-24(i.e. -24<n<0)
94)  52I,  I22,  0,    0        |(Arith & log shift up, n≥ 24; log shift down, n<-24)Exit, ba=(
     I0I,  II9,  II9,  0        |I356 bt'=ba≠h. Set h in bII9
     II3,  I22,  0,    (99)     |I357 bt'=ba≠n. Store ba
     I06,  II9,  0,    (99)     |h or n ≠ ba
     572,  II9,  0,    0        |Set bt and exit
     I2I,  90,   I27,  0        |I362 Fast S/R entry. Set b90'=c+I
     52I,  I27,  II9,  0        |Set bI27 and exit.
     II3,  I22,  0,    (99)     |I364 ba'=(ba & not n)v(bm & n), bII9'=(ba≠bm)& n
     I02,  II9,  I27,  -0.4     |Remove n from BII9, i.e. bII9'=bm
     I06,  II9,  0,    (99)     |≠ba
     I07,  II9,  I27,  -0.4     |& n
     526,  I22,  II9,  0        |≠ ba i.e. ba'=((ba≠bm)& n)≠ba = required result. exit
     52I,  0,    0,    0        |I37I bI2I=Ba, bII9'=N+bm. Dummy B-type extracode.
72)  I2I,  92,   II9,  24       |(I342,3 continued, n<0) Set b92=n+24
     2I6,  I26,  92,   (98)     |Jump if -24 ≤n<0
     I20,  II9,  0,    -23.7    |If n<-24, set bII9=|n| and mark odd,
     I2I,  I26,  0,    (46)     |  and jump to reduce mod 24
     I0I,  II9,  II9,  0        |I376 bt'=ba & h. Set h in bII9
     II3,  I22,  0,    (99)     |I377 bt'=ba & n Store ba
```

```
                                        |I300 Extracodes, Page 2
    107,  119,  0,    (99)              |(I376,7 continued) h or n & ba
    572,  119,  0,    0                 |Set bt and exit
3)  121,  95,   126,  -(0)-1.4          |(I302,3,4,I2,I3,I4) Set mark in B95
    356,  0,    0,    2(99)             |)Preserve A
    357,  0,    0,    3(99)             |)
    113,  119,  0,    0.4(99)           |Store n       )
    216,  126,  119,  2(0)              |Jump if n > 0 )|n| in store
    111,  119,  0,    0.4(99)           |Store -n      )
    127,  119,  0,    *4                |BII9'=*4 if n<0, =0 if n ≥0
    113,  0,    0,    (99)              |Set exponent and top of mantissa for |n|
    113,  122,  0,    1.4(99)           |Store ba
    103,  97,   0,    1.4(99)           |b97'=-ba
    217,  126,  97,   3(0)              |Jump if ba ≥0
    113,  97,   0,    1.4(99)           |Store -ba, i.e. |ba| in store
    126,  119,  0,    0.1*4             |bII9' odd if ba<0, -ve if ba and n are of different s
    113,  0,    0,    1(99)             |Set exponent and top of mantissa for |ba|
    217,  126,  95,   (16)              |Jump if I302,3,I2,I3
    345,  0,    0,    1(99)             |(Here if division (I304,I4)) Set |ba| in L, clear M
    375,  0,    0,    (99)              |Divide by |n|. Result in L, remainder in M
    356,  0,    0,    (99)              |Store remainder
    364,  0,    0,    (0)               |Shift up quotient
    101,  97,   0,    0.4(99)           |Set b97=remainder
    211,  126,  119,  (18)              |Jump to adjust answer if ba>0
    120,  97,   0,    0                 |Set remainder -ve if ba <0
    121,  126,  0,    (18)              |Jump to adjust answer
    163,  92,   0,    0                 |(Reduction loop for I342,3) If M≠0, set b92=16M
    163,  92,   0,    0                 |b92'=8M
    127,  119,  0,    31.1              |Set bII9=m
    124,  119,  92,   0                 |bII9'=8M+m, i.e. have removed 24M
46) 165,  92,   119,  -32               |(Enter here) Regard bII9 as 32M+m. Extract 32M
    215,  126,  92,   -5(0)             |Jump back if M≠0
    122,  119,  0,    24                |If M now = 0, subtract 24 from m.
    216,  126,  119,  -1(0)             |Subtract a further 24 if still +ve
    211,  126,  119,  2(0)              |)
    120,  119,  0,    -23.7             |)Set bII9 = (-24+|n| reduced) if n<0
    121,  126,  119,  (45)              |Jump to shift
97) 163,  92,   0,    0                 |(Log.shift down,n>-24) Halve b92
    107,  91,   92,   0.2*40052115      |Preserve ba where zeros needed
    100,  91,   0,    (99)              |Set zeros at foot of ba
    121,  126,  119,  (45)              |Jump to shift
41) 121,  92,   119,  -24               |(Arith and logical shift up) Set b92=ba-24
    216,  126,  92,   (94)              |Jump if n≥0
    163,  119,  0,    0                 |(ba<0) Halve n as mod for engineers test constants
    107,  91,   119,  *40052115         |Set zeros at top of ba
    121,  126,  92,   (45)              |Jump to shift
40) 113,  122,  0,    (99)              |(I340,1,3,4)) Set b91=ba
    101,  91,   0,    (99)              |           )
    216,  126,  119,  (41)              |Jump if n>0 to shift up
    121,  92,   119,  23.4              |Set b92 =n+23.4
    210,  126,  126,  (44)              |Jump if logical shift down
    216,  126,  91,   (44)              |(Arithmetic shift down) Jump for logical shift if ba
    217,  126,  92,   4(0)              |(Arithmetic shift down, ba<0) Jump if n≤-24
    163,  92,   0,    0                 |Halve b92 as mod for eng. tests consts.
    147,  91,   92,   0.2*40052115      |''or'' ones to foot of ba
    121,  126,  119,  (45)              |Jump to shift
    521,  122,  0,    -0.1              |(Arithmetic shift down ba<0, n<-24) Set ba =-0.1 and
42) 113,  122,  0,    (99)              |(I342,3)) Set b91=ba
    101,  91,   0,    (99)              |        )
    217,  126,  119,  (72)              |Jump if n<0
    122,  119,  0,    24                |(n>0) Subtract 24
    216,  126,  119,  (46)              |Jump to reduce mod 24 if not 0<n<24
98) 121,  126,  119,  (45)              |Jump to shift if 0<n<24 or -24≤n<0
53) 121,  122,  0,    0.1*4             |(I353) Set ba digit 23=1. Interrupt if Ba=126
    113,  122,  0,    3*6               |Inhibit interrupts
                                        |                                    30.8.63
```

```
                                        |I300 Extracodes, Page 3
      I2I,  I23,  II9,  0               |(I353 continued) n -> BI23
      I2I,  I22,  I23,  0               |BI23 -> ba
      5I3,  0,    0,    3*6             |De-inhibit interrupts and exit
43)   I25,  9I,   0,    0               |(Shift table)
      I25,  9I,   0,    0
      I25,  9I,   0,    0
      52I,  I22,  9I,   0               |Exit
      I63,  9I,   0,    0               |(Here for shift up one) Shift down one
      I63,  9I,   0,    0               |(Two) down one
      I63,  9I,   0,    0               |(Three) down one
      I63,  9I,   0,    0               |(Four) down one
      I63,  9I,   0,    0               |(Five) down one
      I2I,  I26,  0,    2(43)           |(Six) Jump to shift up six and exit
      I63,  9I,   0,    0               |(Seven) Shift down one
      I63,  9I,   0,    0               |(Eight) down one
      I63,  9I,   0,    0               |(Nine) down one
      I63,  9I,   0,    0               |(Ten) down one
      I63,  9I,   0,    0               |(Eleven) down one
      I2I,  I26,  0,    I(43)           |(Twelve) Jump to shift up I2 and exit
      I63,  9I,   0,    0               |(I3) Shift down one
      I63,  9I,   0,    0               |(I4) down one
      I63,  9I,   0,    0               |(I5) down one
      I63,  9I,   0,    0               |(I6) down one
      I63,  9I,   0,    0               |(I7) down one
      I2I,  I26,  0,    (43)            |(I8) Jump to shift up I8 and exit
      I63,  9I,   0,    0               |(I9, i.e. down 5) Shift down one
      I63,  9I,   0,    0               |(u.20, i.e.d.4) down one
      I63,  9I,   0,    0               |(u.2I, i.e.d.3) down one
      I63,  9I,   0,    0               |(u.22, i.e.d.2) down one
      I63,  9I,   0,    0               |(u.23, i.e.d.I) down one
45)   52I,  I22,  9I,   0               |(u.24, i.e.d.0) Set b9I in ba and exit
```

```
(0)=I024*4
    I2I,  I26,  0,    (I/I600)      |I400 ca'=log s:
          0     /     0
    I2I,  I26,  0,    (2)           |I402 ca'=exp s:
    324,  0,    II9,  0             |I403 ca'=conj s:    )
    356,  I22,  0,    0             |                    )Transfer real part
257 325,  0,    II9,  I             |(I425 continued joins))
    756,  I22,  0,    I             |                    )Negate and transfer imag. part. Exit
          0     /     0
    I2I,  I26,  0,    (I0/I600)     |I4I0 ca' =sq.rt. s:
    I2I,  I26,  0,    0.I(II/I600)  |I4II am'=arg s:
    I2I,  I26,  0,    0.I(I2/I600)  |I4I2 am'=mod s:
    I2I,  I26,  0,    0.I(2)        |I4I3 ca'=s cos s:, s sin s*
    I2I,  I26,  0,    (I4)          |I4I4 ca'=I/s:
    3I4,  0,    II9,  0             |I4I5 Pseudo-Random Number.  Set s in A
    352,  0,    II9,  I             |Multiply by s*, double-length, non-standardized
    757,  0,    II9,  0.4           |Store l.s.½ in s or s*.  Exit
    I2I,  I26,  0,    (20)          |I420 ca'=ca+s:
    324,  I22,  0,    I             |I42I ca'=ca-s:
    32I,  0,    II9,  I             |Subtract imaginary parts
    I2I,  I26,  0,    (2I)          |Jump to continue
    I2I,  I26,  0,    (24)          |I424 ca'=s:
    325,  0,    II9,  0             |I425 ca'=-s: )
    356,  I22,  0,    0             |             )Negate and transfer real part
    I2I,  I26,  0,    (25)          |Jump to negate and transfer imaginary part and exit
    203,  I26,  II9,  (30)          |I430 s(I)'=s(I)+s(2)   )
    203,  I26,  II9,  0.I(30)       |I43I s(I)'-s(2)        )Jump reducing bII9 by I
    203,  I26,  II9,  0.4(30)       |I432 s(I)'=am.s(2)     )  if n≠0
    203,  I26,  II9,  0.5(30)       |I433 s(I)'=s(I)+am.s(2))
    203,  I26,  II9,  (34)          |I434 s(I)'=s(2)        )
    52I,  0,    0,    0             |Exit if n=0
    203,  I26,  II9,  0.4(34)       |I436 am'=sum s (Ii).s(2i) )Ditto
    203,  I26,  II9,  0.5(34)       |I437 a'=sum s(Ii).s(2i)   )
    52I,  0,    0,    0             |Exit if n=0
    II3,  I22,  II9,  0.4           |I44I sx'=ba, sy'=I2.  Store ba in l.s.½ of s
    I2I,  92,   0,    *03           |Set exponent =I2
    I0I,  9I,   II9,  0.4           |Set ba in B9I
    2I7,  92,   9I,   *03I77777     |Propagate sign digit into m.s.½
    5I3,  92,   II9,  0             |Store exponent and propagated sign digit in l.s.½ of s. Exit
24) 324,  0,    II9,  I             |(I424))
    356,  I22,  0,    I             |       )Transfer imaginary part
    324,  0,    II9,  0             |)Transfer real part and exit
95) 756,  I22,  0,    0             |) (Also end of I400; store am in c(ba) and exit)
    342,  0,    II9,  0             |I452 m'=m.sx times 8 to(ya+ys-ba), ya'=ba (x)
    I2I,  I26,  0,    (47/I700)     |Multiply m by s and jump to set exponent
          0     /     0
          0     /     0
    324,  I22,  0,    0             |I456 s:'=ca )
    356,  0,    II9,  0             |            )Transfer real part
    324,  I22,  0,    I             |)
    756,  0,    II9,  I             |)Transfer imaginary part and exit
    334,  I22,  0,    0             |I462 ca'=ca.s:     )
    362,  0,    II9,  0             |                   )Form product of real parts
    356,  0,    0,    (99)          |Store
    I2I,  I26,  0,    (62)          |Jump to continue
    I2I,  I26,  0,    (66)          |I466 a'=C(s+bm+ba).C(s+bm) + a
    II3,  I22,  0,    I(99)         |I467 am'=Polynomial sum. Store ba =no. of terms
    346,  0,    0,    (99)          |Store am. Set am =0
    I0I,  9I,   0,    I(99)         |Set count in B9I
    I2I,  I26,  0,    (67)          |Jump to polynomial loop
    I2I,  I26,  0,    (4I/I700)     |I473 m'=(xa/xs) times 8 to (ya-ys-ba), ya'=ba (X)
    347,  0,    0     (0)           |I474 C(ba')= quotient (am/s),am'= remainder (X). Clear 1
    I2I,  I26,  0,    (8I/I700)     |I475 C(ba')=quotient (a/s),am'=remainder (X).
    I2I,  I26,  0,    0.I(80/I700)  |I476 C(ba')= quotient) ([am]/s),am'=remainder (X)
    I2I,  I26,  0,    0.4(77/I700)  |I477 Remainder and Adjusted Integral Quotient after division
```

```
I4I     334,  0,    II9,  0          | (I4I4) s
        362,  0,    II9,  0          |s.s
        356,  0,    0,    (99)       |Store
        334,  0,    II9,  I          |s*
        362,  0,    II9,  I          |s*.s*
        320,  0,    0,    (99)       |Add s.s.
        356,  0,    0,    I(99)      |Store
        325,  0,    II9,  I          |-s*
        374,  0,    0,    I(99)      |Divide by (s.s + s*.s*)
        356,  I22,  0,    I          |Store as imaginary part of result
        324,  0,    II9,  0          |s
        374,  0,    0,    I(99)      |Divide by (s.s + s*.s*)
        756,  I22,  0,    0          |Store as real part of result. Exit
20)     324,  I22,  0,    I          |(I420)
        320,  0,    II9,  I          |Add real parts
        356,  I22,  0,    I          |Store
        324,  I22,  0,    0
        320,  0,    II9,  0          |Add imaginary parts
        756,  I22,  0,    0          |Store and exit
I[)     356,  I22,  0,    I          |(I42I continued) Store imaginary part of result
        324,  I22,  0,    0
        32I,  0,    II9,  0          |Subtract real parts
        756,  I22,  0,    0          |Store and exit
62)     334,  I22,  0,    I
        363,  0,    II9,  I          |Negative product of imaginary parts
        320,  0,    0,    (99)       |Add product of real parts
        356,  0,    0,    I(99)      |Store temporarily (=real part of result)
        334,  I22,  0,    0          |)
        362,  0,    II9,  I          |)
        356,  0,    0,    (99)       |) Form sum of 'cross-products'.
        334,  I22,  0,    I          |)
        362,  0,    II9,  0          |)
        320,  0,    0,    (99)       |)
        356,  I22,  0,    I          |Store as imaginary part of result
        334,  0,    0,    I(99)      |)
        756,  I22,  0,    0          |)Extract and store real part and exit
66)     356,  0,    0,    (99)       |(I466) Store am
        355,  0,    0,    (0)        |Shift up 1
        356,  0,    0,    2(99)      |Store
        324,  0,    II9,  0          |C(S+bm)
        342,  I22,  II9,  0          |times C(s+bm+ba)
        I2I,  II9,  0,    0          |Set bII9=0
        I2I,  I26,  0,    (68)       |Jump to add a
30)     I2I,  92,   I26,  -I.4-(0)   |(I430,I,2,3) Set mark in b92
        II3,  I22,  0,    (99)       |Store ba
        I24,  I2I,  0,    0.4        |Step on BI2I to point at Ba*
        I0I,  9I,   0,    (99)       |Set b9I=ba
        2I6,  I26,  92,   (32)       |Jump if I432,3
        2I0,  I26,  92,   (3I)       |Jump if I43I
        324,  9I,   II9,  0          |(I430 continued) sIi
        320,  I22,  II9,  0          |+s2i
        356,  9I,   II9,  0          |Store in sIi
        203,  I26,  II9,  -3(0)      |Cycle, counting
        52I,  0,    0,    0          |Exit
3I)     324,  9I,   II9,  0          |(I43I continued) sIi
        32I,  I22,  II9,  0          |-s2i
        356,  9I,   II9,  0          |Store in sIi
        203,  I26,  II9,  -3(0)      |Cycle, counting
        52I,  0,    0,    0          |Exit
32)     356,  0,    0,    (99)       |(I432,3 continued) Store am
        2II,  I26,  92,   3(0)       |Jump if I432 into loop.
        I2I,  I26,  0,    (33)       |(I433 continued) Jump into loop
        356,  9I,   II9,  I          |(I432 loop) Store in sIi
```

```
    362,  0,    0,    (99)        |(I432 continued) Multiply by ''am''
    203,  I26,  II9,  -3(0)       |Cycle, counting
    756,  9I,   II9,  0           |Store last element and exit
    356,  9I,   II9,  I           |(I432 loop) Store in sIi
33) 324,  I22,  II9,  0           |(Enter here) s2i
    362,  0,    0,    (99)        |Multiply by ''am''
    320,  9I,   II9,  0           |Add sIi
    203,  I26,  II9,  -4(0)       |Cycle, counting
    756,  9I,   II9,  0           |Store last element and exit
34) I2I,  92,   I26,  -I.4-(0)    |(I434,6,7) Set mark in B92
    II3,  I22,  0,    (99)        |Store ba
    I24,  I2I,  0,    0.4         |Step on BI2I to point at Ba*
    I0I,  9I,   0,    (99)        |Set ba in B9I
    2I6,  I26,  92,   (36)        |Jump if I436,7.
    II3,  I22,  0,    I(99)       |(I434 continued) Store ba*
    I02,  9I,   0,    I(99)       |ba-ba* in B9I
    2I7,  I26,  9I,   6(0)        |Jump if ba*>ba, i.e. if transfer backwards
    I0I,  9I,   0,    (99)        |Set ba in B9I
    334,  I22,  II9,  0           |(loop) Extract element from s2 starting at highest add
    356,  9I,   II9,  0           |Store in sI
    203,  I26,  II9,  -2(0)       |Cycle, reducing modifier
    52I,  0,    0,    0           |Exit
    I23,  92,   II9,  0           |(ba*>ba) Set b92=-(n-I)
    I04,  II9,  0,    (99)        |SET ba+(n-I) in BII9
    I20,  9I,   II9,  0           |Set B9I= ba+n-I-(ba-ba*),=ba*+n-I
    334,  9I,   92,   0           |(loop) Extract element from s2, starting at lowest add
    356,  II9,  92,   0           |Store in sI
    20I,  I26,  92,   -2(0)       |Cycle increasing modifier
96) 52I,  0,    0,    0           |Exit
36) 346,  0,    0,    (0)         |(I436,7 contiued) Set zero in A
    2I0,  I26,  92,   (37)        |Jump if I437
    356,  0,    0,    (99)        |Store partial sum (zero initially)
    324,  I22,  II9,  0           |Extract s2i
    362,  9I,   II9,  0           |Multiply by sIi
    320,  0,    0,    (99)        |Add previous partial sum, single length, QR
    203,  I26,  II9,  -4(0)       |Cycle, reducing modifier
    52I,  0,    0,    0           |Exit with result in Am
37) 356,  0,    0,    (99)        |(I437 continued) Store m.s.½ of partial sum (zero ini
    355,  0,    0,    (0)
    356,  0,    0,    2(99)       |Store l.s.½ of partial sum
    324,  I22,  II9,  0           |Extract s2i
    342,  9I,   II9,  0           |Multiply by sIi, double length, Q.
68) 356,  0,    0,    I(99)       |(I466 continued joins) Store m
    355,  0,    0,    (0)         |Shift l into m
    320,  0,    0,    2(99)       |Add c(2(99)) to ''l''
    356,  0,    0,    3(99)       |Store
    324,  0,    0,    (99)        |c((99))
    300,  0,    0,    I(99)       |Add ''m''
    356,  0,    0,    4(99)       |Store
    355,  0,    0,    (0)         |Shift up l
    300,  0,    0,    3(99)       |Add sum of l.s. halves
    3I0,  0,    0,    4(99)       |Add m.s.½ of sum of m.s. halves
    203,  I26,  II9,  (37)        |Cycle, reducing modifier (not I466)
    52I,  0,    0,    0           |Exit
67) 362,  0,    0,    (99)        |Multiply by ''am''
    320,  II9,  9I,   0           |Add coefficient
    203,  I26,  9I,   -2(0)       |Cycle, reducing modifier
    52I,  0,    0,    0           |Exit
2)  324,  0,    II9,  0           |(I402,I3) Set s in Am
    I2I,  97,   0,    *4003630    |Set link
    2II,  I26,  I26,  (36/I500)   |Jump if I402
75) 356,  0,    0,    2(99)       |Preserve s
    324,  0,    II9,  I           |Set s* in Am
    I2I,  I26,  0,    (86/I500)   |Jump to form cos
```

|I400 Extracodes, Page 4

(0)=*4002350

```
50) I02,  II9,  I27,  -0.4      |(II02) Subtract n from bII9 i.e. bII9'=bm
51) I2I,  9I,   I27,  0         |(II00,II0I join) Set b9I'=c+I
    I2I,  I27,  II9,  0,        |Set c'= bII9 (=s,n,bm for II00,II0I,
                                |              II02 respectively).
    52I,  I22,  9I,   0         |Set ba'=c+I (from b9I) and exit
```

(0)=*4002354

```
54) 5I3,  II9,  0,    6*6      |(II24)Set n in V6 and exit
55) I07,  II9,  0,    6*6      |(II25) bII9'=v6 & n
    52I,  I22,  II9,  0        |Copy to ba and exit
57) II3,  I22,  0,    (99)     |(II3I) Store ba
    I0I,  9I,   I27,  0        |Set C(c+I) in B9I
    I65,  92,   9I,   2047     |Set 1 (count) in B92
    I25,  9I,   0,    0        |)Shift k to integer position in B9I
    I25,  9I,   0,    0        |)
    I0I,  93,   0,    (99)     |Set ba in B93
    20I,  I26,  I27,  2(0)     |Jump into loop and increase bI27 by I
    I64,  93,   9I,   5II.4    |Add k to b93
    I0I,  94,   93,   0        |Set C(b93) in B94(first time C(ba), then C(ba+k) etc)
    I07,  94,   I27,  -0.4     |Mask b94 with m (=C(c+I.4))
    I26,  94,   II9,  0        |Non-equivalent with n
    2I4,  I26,  94,   3(0)     |Jump if zero i.e. test successful
    203,  I26,  92,   -5(0)    |Cycle if non-zero, counting from 1 till zero
    I2I,  93,   0,    *4       |If still unsuccessful after 1 cycles, prepare to set ba=*4
    52I,  I22,  93,   0        |Exit with ba=address of successful ½word (*4 otherwise)
76) I2I,  9I,   0,    3        |Error exit (x<0) for LOG. Set marker
    I2I,  I26,  0,    *4006003 |Jump to monitor
```

```
     (0)=I280*4    140 024
     I2I,  I26,  0,    (8)      |I500 a'=a+s:
     I2I,  I26,  0,    (I)      |I501 a'=a-s:
     I2I,  I26,  0,    (2)      |I502 a'=-a+s:
64)  762,  0,    0,    (99)     |(I562) Multiply by n and exit
     I2I,  I26,  0,    (4)      |I504 a'=s:
5)   325,  0,    II9,  I        |I505 a'=-s:
     70I,  0,    II9,  0
7)   362,  0,    II9,  0        |(I542,3) ''1'' xs
41 0 356,  0,    0,    I(99)    |Store
     324,  0,    0,    (99)     |''m''
     362,  0,    II9,  I        |''m''xs*
     320,  0,    0,    I(99)    |Add ''1''xs
     356,  0,    0,    I(99)    |Store
     324,  0,    II9,  0        |s
     342,  0,    0,    (99)     |a'=''m''xs
     I2I,  I26,  0,    (80)     |Jump to add other components and exit
420  I2I,  I26,  0,    I.I(25)  |I520 am'=am+n.
     I2I,  I26,  0,    0.I(25)  |I521 am'=am-n.
22)  I2I,  I26,  0,    (53)     |(I520,I continued) jump to set am=am+c(99) & exit
     734,  0,    0,    (99)     |(I534,5 continued) set am'=c(99), l'=0 (X). exit
     I2I,  I26,  0,    2(0)     |I524 am'=n,l'=0
25)  I20,  II9,  0,    0        |I525 am'=-n,l'=0 (Also I52I,35) Set bII9'=-n
     II3,  II9,  0,    0.4(99)  |(I520,4,34 join) Store ± n
     I2I,  9I,   0,    *03      |) Set up most significant half of (99) to
43 02I7,  9I,   II9,  *03I77777|)      give floating point number from ± n
     II3,  9I,   0,    (99)     |)    with exponent I2 and sign copied up
     2II,  I26,  I26,  (60)     |Jump if bI26 even (I524,5)
     I22,  I26,  0,    9.4      |Jump back to (22) if I520,I and to I(22) if I534,5
     I2I,  I26,  0,    I.5(25)  |I534 am'=n,l'=0(X)
     I2I,  I26,  0,    0.5(25)  |I535 am'=-n,l'0(X)
65)  356,  0,    0,    (99)     |(I565) store m
     355,  0,    0,    (0)      |Shift 1 to m
440  322,  0,    0,    (98)     |m'=''1''
     7II,  0,    0,    (99)     |Subtract ''m''(i.e.a'=-a) and exit
     I2I,  I26,  0,    0.I(8)   |I542 a'=a.s:
     I2I,  I26,  0,    0.I(2)   |I543  a'=-a.s:
I)   356,  0,    0,    (99)     |(I50I) Store m
     355,  0,    0,    (0)      |Shift 1 to m
     32I,  0,    II9,  I        |''1''-s*QR
     356,  0,    0,    I(99)    |Store
450  324,  0,    0,    (99)     |Bring out ''m''
     30I,  0,    II9,  0        |''m''-s in A
     I2I,  I26,  0,    (80)     |Jump to add (''1''-s*) and exit
53)  356,  0,    0,    I(99)    |(I520,I continued) Store am
     324,  0,    0,    (99)     |Set c(99) in A, standardized
     720,  0,    0,    I(99)    |Add 'am' and exit
     356,  0,    II9,  0        |I556 s:'=a.    Store m in s
     355,  0,    0,    (0)      |Shift 1 to m
440  356,  0,    II9,  I        |Store ''1'' in s*
     730,  0,    II9,  0        |Add back '''m'' to restore a,  exit
     I2I,  I26,  0,    0.I(62)  |I562 am'=am.n
66)  237,  I26,  0,    (65)     |(I566) Jump if ax <0
     740,  0,    0,    (0)      |If ax ≥0, standardise and exit
     I2I,  I26,  0,    (65)     |I565 a'=-a
     I2I,  I26,  0,    (66)     |I566 a'=|a|
     334,  0,    II9,  0        |I567 a'=|s:|.  Set am from s
41 6237,  I26,  0,    (5)      |Jump if ax<0 to I505 (a'=-s:)
4)   324,  0,    II9,  I        |(also I504)   Set am from s*
     700,  0,    II9,  0        |Add s (i.e. a'=s:) and exit
98)       *4   /     0          |Floating - point zero
     347,  0,    0,    (0)      |I574 am'=am/n.   Clear l
     I2I,  I26,  0,    (75)     |I575 am'=aq/n.   Jump for I574,5
     356,  0,    0,    (99)     |I576 a'=a/s:.    Store m
     355,  0,    0,    (0)      |Shift 1 to m
```

```
356,  0,   0,   I(99)     |(I576 continued)  Store ''1''
324,  0,   0,   (99)      |Bring back ''m''
374,  0,   II9, 0         |Divide by s, QR,=(am/s)R
356,  0,   0,   2(99)     |Store
343,  0,   II9, 0         |Multiply by 5,=-(am/s)R x s
356,  0,   0,   3(99)     |Store m.s.½
355,  0,   0,   (0)       |)
356,  0,   0,   4(99)     |)Store l.s.½
324,  0,   0,   3(99)     |Bring back m.s.½
300,  0,   0,   (99)      |Add ''m''
320,  0,   0,   4(99)     |Add l.s.½ of -(am/s)R x s
320,  0,   0,   I(99)     |Add ''1''
356,  0,   0,   3(99)     |Store, =(a-(am/s)R x s)
324,  0,   0,   2(99)     |Bring back (am/s)R
363,  0,   II9, I         |Multiply by -s*
320,  0,   0,   3(99)     |Add c(3(99)),=(a-(am/s)R x s)-(am/s)R x s*
374,  0,   II9, 0         |Divide by s
700,  0,   0,   2(99)     |Finally add (am/s)R and exit
340,  0,   0,   (0)       |(I574,5) Standardize.
356,  0,   0,   I(99)     |Store m
355,  0,   0,   (0)       |)
356,  0,   0,   3(99)     |)Store l
I2I,  9I,  0,   *03       |(I562 joins)
II3,  II9, 0,   0.4(99)   |)Set up n floating point in (99)
2I7,  9I,  II9, *03I77777 |)
II3,  9I,  0,   (99)      |)
2I0,  I26, I26, (64)      |Jump if bI26 odd (I562)
324,  0,   0,   (99)      |(I574,5) n standardized in A
I2I,  I26, 0,   6(0)      |Jump
340,  0,   0,   (0)       |(I774,5) Standardize a.
356,  0,   0,   I(99)     |Store m
355,  0,   0,   (0)       |)Store l
356,  0,   0,   3(99)     |)
324,  0,   II9, 0         |s standardised in A
356,  0,   0,   (99)      |(I574,5 rejoin)  Store standardised divisor
334,  0,   0,   I(99)     |Bring back m.s. ½ of dividend
774,  0,   0,   (99)      |Divide and exit
52I,  0,   0,   0         |Dummy exit
356,  0,   0,   (99)      |(I502,I543)  Store m
355,  0,   0,   (0)       |Shift l to m
322,  0,   0,   (98)      |Set m =-''1''
3II,  0,   0,   (99)      |Subtract ''m'' (i.e.a'=-a)
356,  0,   0,   (99)      |(I500,I542 join) Store m
355,  0,   0,   (0)       |Shift l to m
2I0,  I26, I26, (7)       |Jump if bI26 odd (I542,3)
320,  0,   II9, I         |(I500,2) Add s* to ''1''
356,  0,   0,   I(99)     |Store
324,  0,   0,   (99)      |Bring back ''m''
300,  0,   II9, 0         |Add s
356,  0,   0,   (99)      |(I50I joins, I542,3 rejoin)  Store m
355,  0,   0,   (0)       |Shift l to m
300,  0,   0,   I(99)     |Add C(I(99))
7I0,  0,   0,   (99)      |Add ''m'' and exit
I2I,  97,  I26, *0000672  |(I730,I,2,3) Set link for exit to (96/I400)
342,  0,   0,   (97)      |SIN/COS.        Multiply by I/2π
2I7,  I26, I24, 3(0)      |Jump if small (<I/8, i.e.x < π /4)
330,  0,   0,   (96)      |Fix with exponent I3 unless very large
355,  0,   0,   (0)       |Take fractional part i.e. reduce mod 2  (zero if lar
2I0,  I26, 97,  2(0)      |Jump if sin
32I,  0,   0,   (94)      |If cos, subtract -I/4 (i.e. add π /2 to x)
362,  0,   0,   (95)      |x½
2I7,  I26, I24, 4(0)      |Jump if exponent - ve, i.e.<+I/8
32I,  0,   0,   (95)      |Subtract ½ (range-3/8 to I/8 , i.e. -3π /2<x<π /2)
2I7,  I26, I24, 2(0)      |Jump if >-I/8 (x>-π /2)
```

```
   322,  0,    0,    (94)      |(Sin/Cos continued) If between -3/8 and -1/8, add 1/4
   356,  0,    0,    (99)      |Store as y                |(range ± 1/8, or ±π/2 in X)
   362,  0',   0,    (99)      |y squared
   121,  91,   0,    4         |Set count
   346,  0,    0,    1(99)     |Store y squared, clear A
   310,  0,    91,   (92)      |(Loop) Add ith coefficient to a
   342,  0,    0,    1(99)     |Multiply by y squared
   203,  126,  91,   -2(0)     |Cycle, forming polynomial in y squared
   310,  0,    0,    (93)      |Add 0th coefficient, giving sin y/y
   342,  0,    0,    (99)      |Multiply by y
   121,  126,  97,   *7777065  |SIN/COS EXIT
97) *00012137/*14066712        |1/2π
96) *03200000/*00000000        |+0, with exponent 13
95) *00040000/*00000000        |1/2
94) *00160000/*00000000        |-1/4
93) *00414441/*76652103        |)
92) *00726521/*03065616        |)
    *01050632/*74015313        |)Coefficients for sin/cos
    *01354645/*66416023        |)
    *01252005/*01240643        |)
    *01306330/*74163500        |)
91) *76737740/*00000000        |Constant for Log, -(256½) x8 to power -8
26) 121,  97,   0,    *4004100 |(1713) Set link
   235,  126,  0,    (90)      |Jump if a ≠ 0 to form log am, exit to (51)
   325,  0,    119,  0         |(a=0) Set -s in A
   236,  124,  0,    *3        |If s ≤ 0, set exponent ='+192', preparing for E0
   237,  124,  0,    *5        |If s >0, set exponent ='-192', preparing for exp underflow
   121,  126,  0,    (89)      |Jump to tail of Exp to set E0 or EU, exit to (50)
51) 342,  0,    119,  0         |(1713 continues) (Log am in A) Multiply by s
35) 121,  97,   0,    *4004100 |(1702/3 join) EXPONENTIAL Set link to exit to (50)
36) 360,  0,    0,    (0)       |(1402 continued joins) Standardize (=x say)
   121,  91,   0,    *4         |Set *4 in B91
   217,  126,  124,  10(0)      |Jump if exponent negative (x small)
   121,  92,   124,  *774       |Set b92 = exponent -4
   216,  126,  92,   -9(0)      |Jump for out of range if exponent ≥ 4
   342,  0,    0,    (46)       |Multiply by log e to base 8
   330,  0,    0,    (44)       |Add ½ and fix, i.e. (Int.pt)R+ in ½ word position
   356,  0,    0,    (99)       |Store
   113,  91,   0,    0.4(99)    |Store *4 in l.s.½, i.e. clear frac.pt. and add ½
   311,  0,    0,    (99)       |Subtract from a, i.e. result =x log e -(Int.pt)R+
   342,  0,    0,    (45)       |Multiply by log 8 to base e, i.e. unscale remainder
   101,  91,   0,    (99)       |Set (Int.pt)R+ at bottom of B91
   356,  0,    0,    1(99)      |Store a, =z say
   372,  0,    0,    1(99)      |z squared
   356,  0,    0,    2(99)      |Store
   330,  0,    0,    (49)       |Add p
   372,  0,    0,    1(99)      |Multiply by z
   356,  0,    0,    (99)       |Store, =(z squared + p) z, =w say
   334,  0,    0,    (47)       |q
   372,  0,    0,    2(99)      |Multiply by z squared
   121,  92,   0,    2.0        |Set count
   330,  0,    0,    (48)       |Add r
   301,  0,    0,    (99)       |Subtract w
   356,  0,    0,    2(99)      |Store
   334,  0,    0,    (99)       |w
   330,  0,    0,    (99)       |2w
   374,  0,    0,    2(99)      |2w/(qzz+r-w), = exp (z/8) -1 approx
   121,  126,  0,    4(0)       |Jump into loop
   356,  0,    0,    (99)       |Store, = v say) Generate successively
   330,  0,    0,    (43)       |Add 2          ) exp(z/4)-1, exp(z/2)-1
   342,  0,    0,    (99)       |Multiply by v ) exp z-1, keeping accuracy
   125,  91,   0,    0          |)Shift b91 up 5 places each time round the loop
   163,  91,   0,    0          |)i.e. end up with (Int.pt)R+ in exponent position
   203,  126,  92,   -5(0)      |Cycle
```

```
    320,  0,    0,    (42)        |(Exp continued) Add 1, i.e. result = exp z
    124,  124,  91,   *001        |Adjust exponent, adding (Int.pt)R+ +1, result =8 exp aq
89) 365,  0,    0,    (0)         |)Shift down to ensure unstandardized) result = exp aq
    340,  0,    0,    (0)         |)and set EC or EU if appropriate    )
    121,  126,  97,   *7776445    |EXIT FROM EXPONENTIAL
4[illegible] *01017006/*40314262  |)
48) *01217006/*40314334           |)Coefficients for Exponential
47) *00460021/*25613606           |)
46) *00036616/*04734165           |Log e to base 8 ) Constants for Exponential
45) *00220505/*31077170           |Log 8 to base e)
44) *01200000/*40000000           |Constant for Exponent. ½,fixed with point at ½ word posn.
43) *00220000/0                   |+2
42) *00210000/0                   |+1
15) 362,  0,    0,    2(99)       |(1402,13 continued) (cos s* in A) Multiply by s (1413),
    356,  122,  0,    0           |    or exp s (1402) and store as real part of ca.
    324,  0,    119,  1           |Set s* in A
    121,  97,   0,    *40036351   |Set link to exit to 2(0)
    121,  126,  0,    (86)        |Jump to form sin s*, exit to 1(0)
    362,  0,    0,    2(99)       |Multiply by s(1413) or exp s (1402)
    756,  122,  0,    1           |Store as imaginary part of ca and exit
24) 121,  97,   0,    *4004011    |(1700,1) LOG Set link to exit to (50)
90) 237,  126,  0,    (76/1400)   |Jump for monitor if a<0
    360,  0,    0,    (0)         |(1713 continued joins) Standardize
27) 234,  126,  0,    (76/1400)   |(1400 continued joins) Jump for monitor if =0
    121,  91,   124,  *4          |Set b91'= exponent +256
    121,  124,  0,    0           |Set exponent =0
    320,  0,    0,    (41)        |Add p
    356,  0,    0,    1(99)       |Store x+p
    300,  0,    0,    (40)        |Add -2p, = x-p
    374,  0,    0,    1(99)       |Divide by x+p
    121,  92,   0,    6           |Set count
    113,  91,   0,    0.4(99)     |Store exponent +256 in lower half of (99)
    356,  0,    0,    1(99)       |Store (x-p)/(x+p), = z say
    362,  0,    0,    1(99)       |Square
    113,  0,    0,    (99)        |Clear top half of (99), i.e.c(99)=(exp +256) x 8 to power -8
    346,  0,    0,    2(99)       |Store z squared.
    300,  0,    92,   (37)        |(Start of loop) Add coefficient )Form
    342,  0,    0,    2(99)       |Multiply by z squared           )polynomial
    203,  126,  92,   -2(0)       |Cycle                           )in z squared
    300.  0,    0,    (38)        |Add 0th coefficient
    352,  0,    0,    1(99)       |Multiply by z, = (log x+½ log 8) x 8 to power -8
    300,  0,    0,    (99)        |Add(exp +256) x 8 to power -8,double length
    310,  0,    0,    (91)        |Add -(256½) x 8 to power -8
    362,  0,    0,    (39)        |Multiply by ln8 x 8 to power 8; result = log x
    121,  126,  97,   *7776534    |EXIT FROM LOG
41) *00026501/*17146376           |''p''              )
40) *00122575/*41463003           |-2p                )Constants for log
39) *02220505/*31077170           |ln8 x 8 to power 8)
60) 724,  0,    0,    (99)        |(1524,5 continued) Set am'=c(99) Q, and exit
    0     /     0                 |)
    0     /     0                 |)
    0     /     0                 |)
    0     /     0                 |)Spare
    0     /     0                 |)
    0     /     0                 |)
    0     /     0                 |)_____
3[illegible] *76075434/*11670327  |)
37) *76024411/*30520752           |)
    *76014237/*13253256           |)Coefficients For Log.
    *76010630/*11271374           |)
    *75666171/*30127254           |)
    *75661015/*40021262           |)
    *75621422/*02664134           |)
    *76011735/*74545451           |)
```

```
    (0)=I536*4 x4003
    +0/0                                    |Unassigned
    I0I,  98,   II9,  0                     |I60I g'=s    Store m.s. ½
    50I,  99,   II9,  0.4                   |Store l.s. ½ and exit
    +0/0                                    |Unassigned
    I2I,  I26,  0,    0.I(5)                |I604 g'=g+s    Jump with marker
    I2I,  I26,  0,    (5)                   |I605 g'=g+s with end-around-carry
    I2I,  I26,  0,    (6)                   |I606 g' =g ≠s
    I07,  98,   II9,  0                     |I607 g' = g&s    M.s. ½
(9) 507,  99,   II9,  0.4                   |L.s. ½ and exit
    I26,  98,   II9,  0                     |I6II g' = not g.    M.s.½
    526,  99,   II9,  0.4                   |L.s.½ and exit
    II3,  98,   II9,  0                     |I6I3 s'=g    M.s. ½
    5I3,  99,   II9,  0.4                   |L.s. ½ and exit
    II3,  98,   0,    (99)                  |I6I5 am'=g    M.s.½ to store
    II3,  99,   0,    0.4(99)               |L.s.½ to store
    734,  0,    0,    (99)                  |Transfer to A and exit
24) II3,  0,    0,    0.4(99)               |(I624) Clear l.s. ½ of word
    I0I,  9I,   II9,  0                     |Set b9I=h
    II3,  9I,   0,    (99)                  |Store b9I in m.s. ½
    734,  0,    0,    (99)                  |Set am and exit
    I2I,  I26,  0,    (24)                  |I624 am'=h
    +0/0                                    |Unassigned
    I2I,  I26,  0,    (92)                  |I626 h'=am
    +0/0                                    |Unassigned
30) I47,  98,   II9,  0                     |I630 g' = g & (not s)
    I47,  99,   II9,  0.4                   |g'=gvs
6)  I06,  98,   II9,  0                     |(I606 joins)
    506,  99    II9,  0.4                   |g'=g≠ s (=g & not s for I630) . Exit
    +0/0                                    |Unassigned
    356,  0,    0,    (99)                  |I635 g'=am    Store am
    I0I,  98,   0,    (99)                  |M.s. ½ in g
    50I,  99,   0,    0.4(99)               |L.s. ½ and exit
1)  356,  0,    0,    (99)                  |(I626) Store am
    I0I,  92,   0,    0.4(99)               |Extract l.s.½
    2I5,  92,   92,   0.I                   |Set b92 =0.I if l.s. ≠ 0
    I47,  92,   0,    (99)                  |Extract m.s. ½ of g, oring 0.I at bottom if l.s.½ ≠ 0
    5I3,  92,   II9,  0                     |(i.e. Atlas type rounding) Store in h and exit
    +0/0                                    |Unassigned
    I47,  98,   II9,  0                     |I646 g'=gvs    M.s. ½
    547,  99,   II9,  0.4                   |L.s. ½ and exit
    +0/0                                    |Unassigned
    +0/0                                    |Unassigned
    I52,  98,   II9,  0                     |I652 bt' =g - s.  Set bt from difference of m.s. halves
    225,  I26,  0,    3(0)                  |Jump if non-zero to ignore l.s. halves
    I52,  99,   II9,  0.4                   |If zero, set bt from difference of l.s. halves
    224,  I26,  0,    (96)                  |Jump to exit if zero
    I0I,  9I,   0,    6*6                   |Extract V6 (l.s. digit = Bcarry)
    I63,  9I,   0,    0                     |Shift Bcarry to sign position
    572,  9I,   0,    0                     |Set bt from Bcarry and exit
5)  I04,  99,   II9,  0.4                   |(I604,5) Add l.s.½ halves
    I0I,  9I,   0,    6*6                   |Extract V6 (l.s. digit = Bcarry for l.s.½)
    I04,  98,   II9,  0                     |Add m.s. halves
    2I0,  I26,  I26,  6(0)                  |Jump if I604
    I0I,  92,   0,    6*6                   |Extract V6 (l.s. digit = Bcarry for m.s.½)
    I64,  98,   9I,   0.I                   |Add carry from l.s.½ into m.s.½
    I47,  92,   0,    6*6                   |Set digit 23 of B92 =I if Bcarry set by -I(0) or -4(0
    I64,  99,   92,   0.I                   |Add into l.s.½
    I0I,  9I,   0,    6*6                   |Extract V6
    564,  98,   9I,   0.I                   |(I604 rejoins) Add final carry, if any,
73)   *00037777 / *76660000                 |).          |from l.s.½ to m.s. ½, and exit
      *00037756 / *67I42647                 |)Coefficients
75)   *00026501 / *I7I46376                 |)  for square root routine
76)   *0I200000 / *40000000                 |)
77)   *00040000 /      0                   |+½
```

```
10) 165,  93,   119,  -1          |(14I0) Set b93 from b119 removing octal fraction
    121,  97,   0,    -3(47)      |Set link to exit to (47)
12) 324,  0,    118,  1           |(14112 joins, BI26 odd; 1400 cont.) Set a = sx, = v say
    237,  93,   93,   0.1         |Set b93 odd if v<0
14) 356,  0,    0,    (99)        |(17112 joins with BI26 odd) Store v
    362,  0,    0,    (99)        |v squared
    356,  0,    0,    1(99)       |Store
    324,  0,    119,  0           |s
    362,  0,    119,  0           |s squared
    320,  0,    0,    1(99)       |Add v squared                    |for 14112,17110,12
22) 2110, 97,   126,  -3(96)      |(17110,1 joins with BI26 odd) Set link to exit to(96)
    360,  0,    0,    (0)         |SQUARE ROOT  Round single length, = x' say
94) 234,  126,  97,   3           |(17120,1,2,3 continued) Exit if a=0(short cut)
    237,  126,  0,    (45)        |Jump to error exit if a<0
26) 165,  91,   124,  *0001       |(14110 second entry) Least sig. digit of exponent to b9
    356,  0,    0,    (99)        |Store x'
    215,  91,   91,   *0003731O   |)Set 1st approximation to sqrt x' in 1(99), =y0 say
    124,  91,   124,  *00062343   |)≑ 4th root of 1/8, with ½ exp of x', if exp even
    113,  91,   0,    1(99)       |)≑(1/8) to power 3/4, with exp ½(bI24+1), if exp odd
    121,  124,  0,    0           |Force bI24=0, giving a'=x0, say
    300,  0,    0,    (75)        |Add constant
    342,  0,    0,    1(99)       |Multiply by y0 to give linear approximation, =y1 say
    121,  92,   0,    1           |Set count for two cycles of loop
    356,  0,    0,    1(99)       |Store y          )
    334,  0,    0,    (99)        |x                )
    374,  0,    0,    1(99)       |Divide by y      )y(n+1)=½(x/y(n)+y(n))
    300,  0,    0,    1(99)       |Add y            )
    342,  0,    92,   (73)        |Multiply by ½    )
    203,  126,  92,   -5(0)       |Cycle            )
    356,  0,    0,    1(99)       |Store                  )Last iteration
    343,  0,    0,    1(99)       |Multiply by -y,        )y(n+1)=y(n)
    310,  0,    0,    (99)        |Add x double length    )+½(x-y(n)squared)/y(n)
    373,  0,    0,    (76)        |Multiply by -½         )
    374,  0,    0,    1(99)       |Divide by y            )
    302,  0,    0,    1(99)       |Negate and add y, d.l.)
    121,  126,  97,   3           |SQUARE ROOT EXIT to b97+3
    +0/0                          |Spare
45) 121,  91,   0,    2.4         |(Square root error exit, argument negative). Set marker
    121,  126,  0,    187*4001    |Jump to Monitor
47) 235,  126,  0,    3(0)        |(14110 continued, with mod s: in A) Jump if ≠0 (normal)
    356,  122,  0,    0           |)If =0, set ca'=0
    756,  122,  0,    1           |)  and exit
    356,  0,    0,    (99)        |If mod s: ≠ 0, store
    367,  0,    93,   0           ||s|
    320,  0,    0,    (99)        |Add mod s:
    362,  0,    0,    (77)        |x½
    121,  95,   0,    *001        |Set mask
    121,  97,   0,    -3(60)      |Set link to exit to (60)
    121,  126,  0,    (26)        |Jump to form sq.rt(½(mod s: + |s|))
60) 107,  95,   93,   0           |)
    214,  126,  95,   4(0)        |(Jump if s≥0, setting b95=0
    121,  95,   0,    -1          |Set b95=-1 if s<0
    211,  126,  93,   2(0)        |Jump if s*>0
    322,  0,    0,    (36)        |If s and s* both <0, negate accumulator
    356,  0,    0,    (99)        |Store, as z say
    324,  0,    93,   1           |)
    362,  0,    0,    (77)        |)½s*
    374,  0,    0,    (99)        |Divide by z
    356,  122,  95,   1           |Store as real pt. of Ca if s<0, imag pt if ≥0
    120,  95,   0,    0           |Negate b95
    324,  0,    0,    (99)        |z
    756,  122,  95,   0           |Store as real pt. of Ca if s≥0, imag. pt if <0. Exit
    +0/0                          |Spare
    +0/0                          |Spare
```

```
25) 360,  0,   0,   (0)      |(1724/5) Standardize
    121,  96,  0,   0        |Set marker
    121,  97,  0,   (96)     |Set link to exit to Dummy Exit
 3) 234,  126, 0,   (95)     |ARCTAN/COT. Jump to short cut if =0
    121,  92,  124, *-1      |Set b92= exponent minus one
    236,  126, 0,   3(0)     |Jump if a ≥0
    366,  0,   0,   (0)      |Otherwise set positive
    126,  96,  0,   0.5      |and reverse digits 21,22 of B96
    217,  126, 92,  5(0)     |Jump if |x| <1
    356,  0,   0,   (99)     |)Otherwise form
    334,  0,   0,   (94)     |)reciprocal
    374,  0,   0,   (99)     |)and reverse digit 23 of B97
    126,  97,  0,   0.1      |)
    217,  126, 124, 7(0)     |Jump if |x'|<1/8
    330,  0,   0,   (81)     |Add 1/u [u=tan (½(arctan 1/8+π/4))]
    356,  0,   0,   (99)     |Store
    330,  0,   0,   (80)     |Add -(u+1/u), i.e. result = x-u
    372,  0,   0,   (81)     |Multiply by 1/u
    374,  0,   0,   (99)     |Divide by x+1/u. Result = (x-u)/(1-ux)
    121,  92,  0,   0.1      |Mark B92 odd for |x'|≥1/8
    356,  0,   0,   (99)     |Store as y
    342,  0,   0,   (99)     |y squared
    121,  91,  0,   4.0      |Set count
    346,  0,   0,   1(99)    |Store y squared, clear A
    300,  0,   91,  (83)     |(Power series loop) Add coefficient
    342,  0,   0,   1(99)    |Multiply by y squared
    203,  126, 91,  -2(0)    |Cycle
    330,  0,   0,   (82)     |Add first coefficient
    342,  0,   0,   (99)     |Multiply by y
    211,  126, 92,  2(0)     |Jump if x' small
    330,  0,   0,   (84)     |Otherwise add arctan u (approx)
55) 211,  126, 97,  2(0)     |Jump if b97 even
40) 302,  0,   0,   (85)     |If b97 odd (cos, x<1; sin, x>1; tan, x>1),
    300,  0,   96,  0.2(86)  |Add 0 or -π                    |form π/2-result
    211,  126, 96,  2(0)     |Jump if b96 even (1721 to 5,x≥0;1411,1726, s≥0)
    302,  0,   0,   (86)     |Otherwise negate result
    121,  126, 97,  0        |ARC TAN/COT EXIT to b97
11) 334,  0,   119, 1        |(1411,b126 odd, and 1400 continued) Set s* in A
46) 210,  97,  126, (96)     |(1726, B126 odd) Set link to exit
    356,  0,   0,   (99)     |Store a, =x say               |for 1411, 1726
    121,  96,  0,   0        |Clear marker
    324,  0,   119, 0        |s
    234,  126, 0,   8(0)     |Jump if zero
    236,  126, 0,   3(0)     |Jump if >0
    366,  0,   0,   (0)      |)If <0, take modulus
    121,  96,  0,   1.5      |)and set marker
    356,  0,   0,   1(99)    |Store |s|
    324,  0,   0,   (99)     |Bring back x
    374,  0,   0,   1(99)    |Divide by |s|
    121,  126, 0,   (3)      |Jump, form arctan. 1411,1726 exit to (96); 1400 to (
    345,  0,   0,   (99)     |(Here if s=0) Set x in L, sign thro' M, exp unchngd
    234,  126, 97,  0        |Exit if a=0 (i.e. if x=0 also) with result =0
    237,  96,  0,   0.1      |If x≠0, set b96 odd if x < 0
    121,  126, 0,   (40)     |Jump with A effectively containing zero, to form ± π
96) 521,  0,   0,   (0)      |Dummy exit
 1) 121,  97,  0,   (15)     |(1400) Set link
    121,  126, 0,   (12)     |Jump to form sq.rt (s.s + s*.s*) . Exit to 3(15)=(18
71) 235,  126, 0,   (93)     |(1720,1,2,3 continued) Sq.rt (1-x.x) in A. Jump if ≠0
    324,  0,   0,   3(99)    |If zero, recover x (=±1)
    342,  0,   0,   (85)     |Multiply by π/2 (a'=± π/2)
    121,  126, 0,   (95)     |Jump to adjust for sin/cos. Exit to (96)
    +0/0                     |Spare
    +0/0                     |Spare
    +0/0                     |Spare
```

```
     +0/0                                    |Spare
93)  356,   0,    0,    I(99)                |(I720-3 continued) a=Sq.rt(I-xx), (≠0). Store
     324,   0,    0,    3(99)                |x
     374,   0,    0,    I(99)                |Divide by sq.rt(I-xx)
     I2I,   I26,  0,    (3)                  |Jump to form arctan/cot.Exit to (96)
2I)  356,   0,    0,    3(99)                |(I722,3 with 0.7 in BI26;I720,I) Store aq (=x)
     373,   0,    0,    3(99)                |Form -x squared
     I2I,   97,   I26,  -I8                  |Set link (exit to 3(96)=(7I)Form SQ.RT; exit to (96)
     3I0,   0,    0,    (87)                 |Form I-x.x                          |Form arctan/cot)
     I65,   96,   97,   0.2                  |b96'=0(I720,I)or 0.2(I722,3)
     236,   I26,  0,    (94)                 |Jump to form sq.rt(I-x.x), if ≥0. Exit to 3(96)=(7I)
     I2I,   9I,   0,    4                    |IfI-x.x<0, Set mark
     I2I,   I26,  0,    I87*400I             |and jump to Monitor for error.
I5)  356,   I22,  0,    I                    |(I400 continued) a=arctan (s*/s). Store as imag.pt.of
     334,   0,    0,    4(99)                |Bring back sq.rt.(s.s+s*.s*)
     I2I,   I26,  0,    (27/I500)            |Jump to form log. Exit to (95/I400)
I8)  356,   0,    0,    4(99)                |(I400 continued) a=sq.rt(ss+s*.s*). Store
     I2I,   I26,  0,    (II)                 |Jump to form arctan(s*/s). Exit to (I5)
80)  *00353565/*6753II22                     |)
8I)  *00220266/*6574I5II                     |)
82)  *00077777/*7777773I                     |)
83)  *00I52525/*25332676                     |)
     *000I463I/*34747I75                     |) Coefficients for Arctan/Cot
     *00I66674/*23667077                     |)
     *77667345/*25I0037I                     |)
     *7773503I/*054I0443                     |)
84)  *0003507I/*3I247463                     |)
85)  *002I444I/*76652I04                     |  π/2
86)  *40000000/*00000000                     |Floating-point zero
     *00346674/*02253570                     |-π
87)  *002I0000/*00000000                     |+I
```

30.8.63

```
(0)=I792*4
    324,  0,    II9,  0          |I700 am'=log s.  Set aq'=s
    I2I,  I26,  0,    (24/I500)  |I70I am'=log aq. Jump I700,I
    324,  0,    II9,  0          |I702 am'=exp s.  Set aq'=s
    I2I,  I26,  0,    (35/I500)  |I703 am'=exp aq  Jump I702,3
    334,  0,    II9,  0          |I704 a'=int.pt.s.Set a'=s
    I2I,  I26,  0,    (5)        |I705 a'=int.pt.a.Jump I704,5
    334,  0,    II9,  0          |I706 a'=sign s   Set a'=s
    I2I,  I26,  0,    (7)        |I707 a'=sign a   Jump I706,7
    324,  0,    II9,  0          |I7I0 am'=sq.rt.s Set aq'=s
    I2I,  I26,  0,    0.I(22/I600)|I7II am'=sq.rt.aq. Jump I7I0,I
    I2I,  I26,  0,    0.I(I4/I600)|I7I2 am'=sq.rt.(aq.aq+s.s). Jump
    I2I,  I26,  0,    (26/I500)  |I7I3 am'=am to power s. Jump
    I2I,  I26,  0,    (I4)       |I7I4 am'=I/s.  Jump
    356,  0,    0,    (99)       |I7I5 am'=I/am. Store am
    325,  0,    0,    (96)       |Set I in A
    774,  0,    0,    (99)       |Divide by''am'' and exit
    324,  0,    II9,  0          |I720 am'=arcsin s. Set aq'=s
    I2I,  I26,  0,    (2I/I600)  |I72I am'=arcsin aq. Jump I720,I
    324,  0,    II9,  0          |I722 am'=arccos s.  Set aq'=s
    I2I,  I26,  0,    0.7(2I/I600)|I723 am'=arccos aq. Jump I722,3
    324,  0,    II9,  0          |I724 am'=arctan s. Set aq'=s
    I2I,  I26,  0,    (25/I600)  |I725 am'=arctan aq.Jump I724,5
    I2I,  I26,  0,    0.I(46/I600)|I726 am'=arctan(aq/s)
    I2I,  I26,  0,    (27)       |I727 c'=c+I,2 or 3 as am >,=,< s
    324,  0,    II9,  0          |I730 am'=sin s Set aq'=s
    I2I,  I26,  0,    0.I(3I/I500)|I73I am'=sin aq.  Jump I730,I
    324,  0,    II9,  0          |I732 am'=cos s.  Set aq'=s
    I2I,  I26,  0,    (3I/I500)  |I733 am'=cos aq. Jump I732,3
    324,  0,    II9,  0          |I734 am'=tan s.  Set aq'=s
    I2I,  I26,  0,    *4004400   |I735 am'=tan aq  Jump I734,5
    I24,  I26,  0,    0.5        |I736 c'=c+2 if |am| ≥ s. Add 0.5 to bI26
    356,  0,    0,    (99)       |I737 c'=c+2 if |am| < s. Store am
    366,  0,    0,    (0)        |Form |am|
    32I,  0,    II9,  0          |Subtract s
    237,  I26,  I26,  I.3        |Jump if |am|-s<0, to 2(0) if I737, to 3(0) if I736
    2II,  I26,  I26,  2(0)       |(|am|-s>0) Jump if I737
    I24,  I27,  0,    I          |(I736, |am|-s≥0; I737, |am|-s<0) Set c'=c+2
    734,  0,    0,    (99)       |Recover am and exit
94) *064  /     0                |+0, exponent 26
96) *00I  /     0                |Floating-point -I
98) *4    /     0                |Floating-point zero
93) *0004 /     0                |Floating-point +½
    I2I,  I26,  0,    (52)       |I752 m'=ax, exp I2; ay'=ay -I2
    I2I,  I26,  0,    (53)       |I753 ax'=m, exp I2; ay = ay +I2
    2II,  I26,  I24,  (54)       |I754 Round am by R+, Q. Jump when A free
    I2I,  I26,  0,    (55)       |I755 ax'=ax, exp (ay-n); ay'=n
    I2I,  I26,  0,    (56)       |I756 s'=am,am'=s
    I2I,  I26,  0,    (57)       |I757 am'=s/am
    356,  0,    0,    (99)       |I760 am'=am squared
    762,  0,    0,    (99)
    I2I,  I26,  0,    (62)       |I762 m'=ax, exp I2
    I2I,  I26,  0,    (63)       |I763 ax'=m, exp -I2
    I20,  II9,  0,    0          |I764 ax'=ax, exp n.  Set bII9=-n
    I2I,  I26,  0,    (65)       |I765 ax'=ax, exp -n. Jump I764,5
    334,  0,    II9,  0          |I766 am'=|s|, X. Set s in am
    236,  I26,  0,    (97)       |I767 am'=|am|,X. Jump if a ≥ 0 to exit
    732,  0,    0,    (98)       |Set a'=-am+0, ie. negate am, and exit
97) 52I,  0,    0,    0          |I77I bI2I'=Ba, bII9'=N+ba+bm. Dummy A-type extracode
    I2I,  I26,  0,    (72)       |I772 m'=(m.sx), exp I2; ay'=ay+sy-I2
    I2I,  I26,  0,    (73)       |I773 m'=(ax/sx), exp (ay-sy-I2); ay'=I2
    347,  0,    0,    (0)        |I774 am'=am/s. Clear 1
    I2I,  I26,  0,    (74/I500)  |I775 am'=a/s.  Jump I774,5
    I2I,  I2I,  0,    0          |I776 Remainder and quotient. Set bI2I=0
    I2I,  I26,  0,    (76)       |Jump
```

```
5) 217,  124,  124,  0      |(1704,5) Set exponent = 0 if negative
   710,  0,    0,    (94)   |Add 0 with exp 26 (ie shift int.pt. to bottom of L). Exit
27) 356,  0,    0,    (99)         |(1727) Store am
   321,  0,    119,  0      |am-s
   234,  127,  127,  1      |Add 1 to b127 if am=s
   237,  127,  127,  2      |Add 2 to b127 if am<s
   734,  0,    0,    (99)   |Restore am and exit
14) 325,  0,    0,    (96)   |(1714) Set +1 in A
   774,  0,    119,  0      |Divide by s and exit
57) 356,  0,    0,    (99)   |(1757) Store am
   324,  0,    119,  0      |Bring out s
   774,  0,    0,    (99)   |Divide by am and exit
54) 101,  91,   0,    6*6    |(1754) Extract V6
   354,  0,    0,    (0)    |R+
   300,  0,    0,    (98)   |Add zero and standardize, i.e. shift down
   513,  91,   0,    6*6    |Restore V6 and exit            |if result superstandard
53) 124,  124,  0,    *014   |(1753)  Add 12 to exponent
63) 356,  0,    0,    (99)   |(1763 joins) Store am
   345,  0,    0,    (99)   |Set in L
   764,  0,    0,    (0)    |Shift up one octal place and exit
56) 356,  0,    0,    (99)   |(1756) Store am
   334,  0,    119,  0      |Set s in A
   356,  0,    0,    1(99)  |Store
   334,  0,    0,    (99)   |Recover am
   356,  0,    119,  0      |Store in s
   734,  0,    0,    (99)   |Reset s in A and exit
72) 352,  0,    119,  0      |(1772) Multiply a by s
52) 122,  124,  0,    *014   |(1752 joins) Subtract 12 from exponent
62) 121,  91,   124,  1      |(1762 joins) Preserve exponent in B91. Also set d20=1
   121,  124,  0,    *014   |Set exponent =12
   121,  126,  0,    (71)   |Jump
73) 121,  92,   0,    *014   |(1773) Set b92=12 in exponent position
   121,  121,  0,    46     |Set B121 to point at B92
41) 340,  0,    0,    (0)    |(1473 joins)  Standardize
   356,  0,    0,    1(99)  |Store am
   324,  0,    119,  0      |s, standardized
   356,  0,    0,    2(99)  |Store
   324,  0,    0,    1(99)  |Bring back am
   374,  0,    0,    2(99)  |Divide by s
47) 113,  122,  0,    (99)   |(1452 continued joins) Store ba (=b92 =12 if 1773)
   101,  119,  0,    (99)   |Set into B119
55) 122,  119,  124,  *4     |(1755 joins) Set b119'=b119-b124+256 in exponent position
   124,  124,  119,  *4     |Set original b119 in B124
   125,  119,  0,    0      |)Shift b119 to integer position and subtract 256,
   125,  119,  0,    -256   |)i.e. original b119-b124 in integer posn with sign propagated
65) 121,  91,   124,  1      |(1764,5 joins) FIXING ROUTINE.Preserve exp in B91, set d20=1
   217,  126,  119,  5.1(0) |Jump if b119 <0, i.e. shift up required, set marker in B126
   214,  126,  119,  (97)   |Jump to exit if b119=0
   120,  119,  0,    1      |(Shift down) Negate and add 1
   365,  0,    0,    (0)    |Shift down one (ensures correct handling of superstandard nos.)
   214,  126,  119,  (97)   |Jump to exit if b119 now =0, i.e. one shift only was required
   121,  92,   119,  27     |(Shift up rejoins)) Set b119=-27 if b119 <-27
   217,  119,  92,   -27    |                    )i.e. if out of range
   125,  119,  0,    0      |)Shift b119 to exponent position
   125,  119,  0,    0      |)
   211,  126,  126,  6(0)   |Jump if shift down
   123,  124,  119,  *777   |SHIFT UP. set b119 +vely in B124, correcting for 7777 at bottom
71) 340,  0,    0,    (0)    |(1752,62,72 cont join) Standardize i.e. shift up adjusting b124
   217,  126,  124,  4(0)   |Jump if exponent now -ve. i.e. shifted too far
   203,  126,  124,  8(0)   |Jump if exponent >0, i.e. more shift up reqd. Subtract 1
   521,  124,  91,   0      |If exp = 0(i.e. correctly shifted) recover original exp & exit
   121,  124,  119,  0      |SHIFT DOWN. Set b119 (negative) in B124
   310,  0,    0,    (93)   |Add ½ with exponent zero, i.e. shift down correctly and add ½
   357,  0,    0,    (99)   |Preserve l.s. ½
```

30.8.63

```
     331,  0,    0,    (93)     |(Shift down cont) Remove ½ from top(no shifting) and clear L
     344.  0,    0,    (99)     |Recover l.s. ½
     521,  124,  91,   0        |Recover original exponent and exit
     364,  0,    0,    (0)      |(Here if shift up beyond standard required) Shift up
     203,  126,  124,  -1(0)    |Cycle counting
     147,  91,   0,    6*6      |)Set A0 by 'or-ing',
     121,  124,  91.   0        |)          recover original exponent
     513,  91,   0,    6*6      |)          and exit
7)   237,  126,  0,    3(0)     |(1706,7)  Jump if -ve
     234,  126,  0,    2.4(0)   |Jump if zero
     725,  0,    0,    (96)     |If positive set +1 in A and exit
     734,  0,    126,  -100.4   |Set -1 or 0 in A if -ve or zero, Exit
77)  356,  0,    0,    2(99)    |(1477 with 0.4 in B126) REMAINDER.Store quotient
     300,  0,    0,    (94)     |Take integer part,= Q SAY
76)  356,  0,    119,  0        |(1776 joins with b121=0) store Q
     342,  0,    0,    (99)     |x denominator (s)                          )
     356,  0,    0,    4(99)    |Store m.s.½                                )
     355,  0,    0,    (0)      |Shift 1 to m                               )Form a-Qs
     302,  0,    0,    3(99)    |Negate, add l.s ½ , of numerator (a)       ) (=R say)
     356,  0,    0,    3(99)    |Store                                      )
     334,  0,    0,    4(99)    |Bring back Q.s (m.s.½)                     )
     302,  0,    0,    1(99)    |m.s.½ of a-Q.s.                            )
     310,  0,    0,    3(99)    |Add l.s.½ of ditto                         )
     214,  126,  121,  9(0)     |Jump to exit if 1477 Ba=0 or 1776
     356,  0,    0,    4(99)    |Store Rm
     234,  126,  0,    7(0)     |Jump to exit if R=0
     113,  0,    0,    3(99)    |Clear store line
     314.  0,    121,  (99)     |Read denominator, numerator, quotient or zero
     237,  121,  121,  0.4      |Change d21 if  <0 ) Set d21 of B12151
     314,  0,    0.    4(99)    |Recover Rm        )   if remainder not
     237,  121,  121,  0.4      |Change d21 if <0  )   of required sign
     164,  126,  121,  0.4      |Skip if remainder wrong sign
     521,  0,    0,    0        |Exit if remainder correct sign
     311,  0,    0,    (99)     |If remainder wrong sign, form R-s, = a-(Q+1)s
     356,  0,    0,    4(99)    |Store (R-s)m
     314,  0,    0,    (92)     |Set +1 in Am
     357,  0,    0,    3(99)    |Store (R-s)1
     320,  0,    119,  0        |Add 1 to Q (i.e. adjust )
     344,  0,    0,    3(99)    |Recover (R-s)1 in L
     356,  0,    119,  0        |Store adjusted Q
     714,  0,    0,    4(99)    |Recover (R-s)m and exit
92)  *0021 /     0              |Floating-point +1
80)  330,  0,    0,    (94)     |(1476,B126 odd) FIXED PT. DIVISION Take int.pt of am (exp=26)
     124,  124,  0,    *776     |Subtract 2 from exponent correcting for 1(0) and 19(0)
     364,  0,    0,    (0)      |Shift up  a  so that binary point is 3 places from foot of L
81)  121,  93,   0,    -2       |(1474,5 join) Set mask
     123,  91,   124,  *7461    |Set b91 = 25 - exponent, in exponent position, plus *0007
     236,  126,  126,  5        |Jump if a>0, preserving marker in B126
     356,  0,    0,    (99)     |OTHERWISE NEGATE A.  I.e.store m
     355,  0,    0,    (0)      |          shift up 1
     302,  0,    0,    (98)     |          negate
     331,  0,    0,    (99)     |          and add back m negatively
     124,  93,   0,    0.1*4    |    also set marks in B93 (positive, odd)
     375,  0,    119,  0        |Divide by s, quotient in L remainder in M
     121,  92,   0,    -4       |Set mask
     101,  94,   119,  0        |m.s.½ of s                          ) Set b94=0 if
     127,  94,   0,    *00077   |mantissa part except sign digit) mantissa of s =0 or -1.0,
     147,  94,   119,  0.4      |'or' with rest of mantissa      ) set b94 ≠0 otherwise
     214,  126,  94,   (21)     |Jump if mantissa =0 or -1.0 to set D0
     211,  126,  126,  7(0)     |Jump if 1474,5
     356,  0,    0,    (99)     |(1476 continues) Store remainder (R)
     357,  0,    0,    1(99)    |Store quotient Q
     364,  0,    0,    (0)      |Shift up Q one octal place
     107,  91,   0,    1(99)    |3 m.s. bits of Q (mantissa), and reduce exp part to 0
```

|1700 Extracodes, Page 4

```
    215,  126,  91,   (21)    |(1476 continued) Jump for DO if Q too l-*s-
    314,  0,    0,    (99)    |Recover R, leaving Q (shifted up) in L
    347,  122,  0,    0       |(1474,5) Store Q, clearing L
    236,  126,  0,    3(0)    |Jump if R>0
    121,  124,  0,    0       |)If R<0, add 1 (fixed point)
    331,  0,    0,    (96)    |)  to adjust for error due to 375
    104,  92,   0,    6*6     |Clear Bc.  Set b92>0 and reset Bc if |xa|≥|xs|
    164,  93,   0,    6*6     |)Jump to DO if |xa|≥|xs|; otherwise add Qs digit to b93
    216,  126,  92,   (21)    |)i.e.set sign of b93  to ≠ of Q and a
    211,  126,  93,   2(0)    |)Negate R if a<0          |i.e. to not sign of final Q
    332,  0,    0,    (98)    |)  giving true R
    217,  126,  93,   5(0)    |Jump if final Q>0
    356,  0,    0,    (99)    |Store true R,
    335,  122,  0,    0       |)set -Q as final Q in C(ba)
    356,  122,  0,    0       |)
    334,  0,    0,    (99)    |and reset true R in A
    543,  124,  91,   *7631   |Reset exponent for R and exit
21) 374,  0,    0,    (98)    |Cause DO and monitor exit


    (0)=*4004400

    |TAN
    |If x = ½π(n+θ), where -½ ≤ θ <½
    |then tan x = tan (½πθ)= p(θ)/(1-θ.θ) if n even
    |            = -cot(½πθ)= -(1-θ.θ)/p(θ) if n odd

    362,  0,    0,    (88)    |Multiply x by 2/π
    121,  91,   0,    0.1     |Set marker
    217,  126,  124,  (82)    |Jump if small (<1/8)
    320,  0,    0,    (85)    |Add -½
    330,  0,    0,    (86)    |''Fix'', i.e. int.pt. in M, frac.pt. in L
    356,  0,    0,    (99)    |Store int.pt.,= n-1
    355,  0,    0,    (0)     |Set frac.pt.in M
    107,  91,   0,    0.4(99) |Set b91= 0.1 if n-1 odd, 0 otherwise
    300,  0,    0,    (85)    |Add -½ to frac.pt. Result = θ
82) 356,  0,    0,    (99)    |Store θ
    342,  0,    0,    (99)    |Form θ.θ
    121,  92,   0,    3       |Set counter
    356,  0,    0,    1(99)   |Store θ.θ
    330,  0,    0,    (87)    |Add -1. Result = -(1-θ.θ)
    346,  0,    0,    2(99)   |Store -(1-θ.θ). Clear A
    300,  0,    92,   1(89)   |Add coefficient ) Form polynomial
    372,  0,    0,    1(99)   |Multiply by θ.θ )  in  θ.θ
    203,  126,  92,   -2(0)   |Cycle           )
    300,  0,    0,    (89)    |Add 0th coefficient.
    363,  0,    0,    (99)    |Multiply by -θ.θ Result = -p(θ)
    210,  126,  91,   (83)    |Jump if n even
    356,  0,    0,    1(99)   |If n odd, store -p(θ) in 1(99)
    325,  0,    0,    2(99)   |and set+(1-θ.θ) in A
83) 774,  0,    91,   1.7(99) |Divide by c(1(99)) extra) if n odd, result = (1-θ.θ)/-p(θ)
85) *0014/0                   |-½ | by c(2(99)) if n odd, result = -p(θ)/-(1-θ.θ). Exit
86) *032/0                    |0 with exp 13
87) *001/0                    |-1
88) *00050574/*60333447       |2/π
89) *00214441/*76652102       |)
    *00156116/*03120022       |)
    *77767277/*63661370       |) Coefficients for p(θ)
    *77312142/*24070717       |)
    *77116451/*75471372       |)
```

30.8.63.