Division

2022-02-16 Eiiti Wada

This subroutine was made for use with *e1000* and *pi1000*, so no error messages will be issued.

Divides C(0πH)×2^{34}+C(2πH) by C(4πH) by means of non-restoring division, places the quotient in 0πH and remainder in 2πH. Both the dividend and divisor must be positive, and quotient and remainder are also positive. When C(0πH)≥C(4πH), incorrect values may be returned.

		G		K	
	0	A	3	F	
	1	T	45	θ	plant link
	2	A		πH	upper dividend
	3	S	4	πH	− divisor
	4	T		πH	
	5	S	46	θ	P34F
32→	6	T	47	θ	counter
	7	A	2	πH	lower dividend
	8	R		F	⎫ shift right 34 bits
	9	R		F	⎪
	10	R	4	F	⎭
	11	A		πH	upper dividend
	12	G	22	θ	jump if negative
	13	L		D	
	14	S	4	πH	− divisor
	15	U		πH	
	16	S		πH	
	17	L		F	⎫ shift left 34 bits
	18	L		F	⎪
	19	L	64	F	⎭
	20	A	48	πθ	+ long 1 to quotient
	21	E	29	θ	
12→	22	L		D	
	23	A	4	πH	+ divisor
	24	U		πH	
	25	S		πH	
	26	L		F	⎫ shift left 34 bits
	27	L		F	⎪
	28	L	64	F	⎭
21→	29	T	2	πH	
	30	A	47	θ	counter
	31	A	2	F	+1
	32	G	6	θ	jump back if negative
	33	A		πH	final adjustment
	34	G	40	θ	
	35	T		F	
	36	A	2	πH	
	37	L		D	
	38	A	48	πθ	+ long 1 to quotient
	39	E	44	θ	unconditional jump
34→	40	A	4	πH	+ divisor
	41	T		πH	
	42	A	2	πH	
	43	L		D	
39→	44	T	2	πH	
	45	(Z		F)	return
	46	P	34	F	constant
	47	(P		F)	counter
	48	P		D	⎫ long 1
	49	P		F	⎭

An example of non-restoring division

```
                0 1 1 0 1 0
            ---------------
0 0 1 1 1 1 )0 0 1 1 0 0 1 0 0 0 0      d ← d × 2^5, q ← 0, n ← n − d
          - 0 0 1 1 1 1
            -----------
            1 1 1 1 0 1 1               n < 0 → d ← d/2, n ← n + d, q ← q × 2 + 0
            + 0 0 1 1 1 1
              ---------
              0 0 1 0 1 0 0             n ≥ 0 → d ← d/2, n ← n − d, q ← q × 2 + 1
              - 0 0 1 1 1 1
                ---------
                0 0 0 1 0 1 0           n ≥ 0 → d ← d/2, n ← n − d, q ← q × 2 + 1
                - 0 0 1 1 1 1
                  ---------
                  1 1 1 0 1 1 0         n < 0 → d ← d/2, n ← n + d, q ← q × 2 + 0
                  + 0 0 1 1 1 1
                    ---------
                    0 0 0 1 0 1 0       n ≥ 0 → d ← d/2, n ← n − d, q ← q × 2 + 1
                    - 0 0 1 1 1 1
                      ---------
                      1 1 1 0 1 1       n < 0 → n ← n + d, q ← q × 2 + 0
                    + 0 0 1 1 1 1
                      ---------
                      0 0 1 0 1 0
```

```
div(n, d){
    q ← 0; d ← d × 2^5;
    n ← n − d;
    for(i = 0; i < 5; i + +){
        d ← d/2;
        if(n < 0){
            n ← n + d; q ← 2 × q;}
        else{
            n ← n − d; q ← 2 × q + 1;}}
    if(n < 0){
        n ← n + d; q ← 2 × q;}
    else{
        q ← 2 × q + 1;}
    (q, n)}
```

e1000

Calculates the Euler's or Napier's constant $e = 2.7182818284\ldots$ up to 1000 places by

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!} + \cdots$$

until $n = 460$.

```
      PKT45K
   45 P    56 F  H for division subroutine
   46 P    64 F  N for main program
   47 P    80 F  M for constants
      T    56 K
  0H (PFPF    )  upper dividend, remainder
  2H (PFPF    )  lower dividend, quotient
  4H (PFPF    )  divisor
      T    64 K
  0N PDPF       long 1
  2N (P230FPF )  460 n (†P128FPF,P62FPF,P36FPF)
  4N (P229DPF )  459 n−1 (†P127DPF,P61DPF,P35DPF)
  6N (P229FPF )  458 n−2 (†P127FPF,P61FPF,P35FPF)
  8N (P228DPF )  457 n−3 (†P126DPF,P60DPF,P34DPF)
 10N (PFPF    )  first divisor 11048217210
 12N (PFPF    )  first dividend 11072445640
 14N (PFPF    )  work space (ws)
      T    80 K
  0M T  1000 D  (†T900D,T842D,T822D)
  1M P     2 F
  2M A   800 D
  3M π       F
  4M M       F
  5M S  1000 D  (†S900D,S842D,S822D)
      T   100 K
```

division subroutine

```
      T   200 K
      G       K
6 → 0 T       D ┐ clear 800D...998D
    1 (T  800 D)│
    2 A     1 θ │
    3 A     1 M │ P2F
    4 U     1 θ │
    5 S       M │ T1000D
    6 G       θ ┘
    7 T       D
    8 H     8πN   n−3
    9 V     6πN   × (n−2)
   10 L       F ┐ left shift 33 bits
   11 L       F ┘
   12 L    32 F
   13 T       D   (n−3)(n−2)/2
   14 H       D
   15 V     4πN   × (n−1)
   16 L       F ┐ left shift 34 bits
   17 L       F ┘
   18 L    64 F
   19 T       D   (n−3)(n−2)(n−1)/2
   20 H       D
   21 V     2πN   ×n
   22 L       F ┐ left shift 33 bits
108→23 L      F ┘
   24 L    32 F   (n−3)(n−2)(n−1)n/4
   25 T   10πN   denominator
   26 A    8πN   (n−3)
   27 A     πN   +1
   28 T       D
   29 H       D
```

```
30   V   6 πN   × (n−2)
31   L     F   ┐ left shift 34 bits
32   L     F   │
33   L  64 F   ┘
34   A    πN   +1
35   T     D
36   H     D
37   V   4 πN   × (n−1)
38   L     F   ┐ left shift 34 bits
39   L     F   │
40   L  64 F   ┘
41   A    πN   +1
42   T     D
43   H     D
44   V   2 πN   ×n
45   L     F   ┐ left shift 32 bits
46   L     F   │
47   L  16 F   ┘ n(1+(n−1)(1+(n−2)(1+(n−3))))/4
48   T  12 πN   numerator
49   T     F
50   T  14 πN   ws, divide by 4 loop
51   A   2 M    A800D
52 U  60 θ      ← 71
53   A   3 M    πF π=U−A
54   U  63 θ    U800D
55   A 110 θ    TF T=S−U
56   U  64 θ    S800D
57   S   5 M    S1000D
58   E  72 θ    exit loop
59   T     D
60  (A 800 D)
61   R   1 F    right shift 2 bits
62   A  14 πN   ws
63  (U 800 D)
64  (S 800 D)
65   L     F   ┐ left shift 34 bits
66   L     F   │
67   L  64 F   ┘
68   T  14 πN   ws
69   A  60 θ
70   A   1 M    P2F
71   G  52 θ    loop back
72 T     D      ← 58
73   T  14 πN   ws
74   A  10 πN
75   T   4 πH   divisor
76   A  12 πN
77   T 800 D
78   A   2 M
79 U  87 θ      ← 97
80   S   4 M    MF A−T=M
81   U  92 θ
82   S     M    T1000D
83   E  98 θ
84   T     F
85   A  14 πN   previous remainder
86   T    πH    to upper dividend
87  (A 800 D)   lower dividend
88   T   2 πH
89   A  89 θ
90   G 100 F    division subroutine
91   A   2 πH   quotient
92  (T 800 D)   store quotient
93   A    πH    remainder
94   T  14 πN   store remainder to ws
95   A  87 θ    advance address
96   A   1 M
97   G  79 θ    loop back
98 T     F      ← 83
99   A   8 πN
```

```
100 |  S      πN  |  −1
101 |  U    2πN  |  n'
102 |  S      πN  |  −1
103 |  U    4πN  |  n'−1
104 |  S      πN  |  −1
105 |  U    6πN  |  n'−2
106 |  S      πN  |  −1
107 |  U    8πN  |  n'−3
108 |  E    7  θ  |
109 |  H    8  F  |  restore multiplication register
110 ‖ T       F  |  used as a constant
111 |  E   34  F  |  return to the initial orders
    E200KPF       start program
```

long fraction print routine

$$e \approx 1 + 1/1! + 1/2! + 1/3! + \cdots + 1/n! \qquad n \equiv 0 \pmod 4$$
$$= 1 + 1/1(1 + 1/2(1 + 1/3(1 + \cdots + 1/n)\cdots))).$$

Let's denote:
$$Q_{n-4} = 1 + 1/(n-3)(1 + 1/(n-2)(1 + 1/(n-1)(1 + 1/n \cdot Q_n)))$$
$$= (n(1 + (n-1)(1 + (n-2)(1 + (n-3)))) + Q_n)/(n(n-1)(n-2)(n-3)).$$

When $Q_{460} = 1$, $Q_0 \approx e$.

pi1000

Calculates the Ludolf's constant $\pi = 3.1415926535\ldots$ up to 1000 places by

$$\frac{\pi}{4} = 4\tan^{-1}\frac{1}{5} - \tan^{-1}\frac{1}{239}, \qquad \text{where} \quad \tan^{-1}\frac{1}{z} = \frac{1}{1\cdot z^1} - \frac{1}{3\cdot z^3} + \frac{1}{5\cdot z^5} - \cdots + \frac{(-1)^n}{(2n+1)\cdot z^{2n+1}} + \cdots$$

until $n = 1433$ for $\tan^{-1}(1/5)$ and $n = 423$ for $\tan^{-1}(1/239)$.

```
      PKT45K
   45 P    56 F  H for divison
   46 P    64 F  N for main program
   47 P   200 F  M for subroutine
      T    56 K
  0H (PFPF    )  upper dividend, remainder
  2H (PFPF    )  lower dividend, quotient
  4H (PFPF    )  divisor
      T    64 K
  0N (P40FPF  )  80←956
  2N (P12DPF  )  25←57121
  4N (P717DPF )  1435 last i = k ←425(†P363DPF,P147DPF,P75DPF)
  6N (P478FPF )  956=239×4
  8N (P28560DPF) 57121=239×239
 10N (P212DPF )  425 last i = k(†P107DPF,P42DPF,P23DPF)
 12N PDPF        long 1
 14N P1FPF       long 2
 16N PFP32768F   2^34
 18N (PFPF    )  i
 20N (PFPF    )  work space, ws
 22N S    2πH   switch constant
 23N S   12πN   switch constant
 24N T   600 D
 25N T   800 D  (†T702D,T642D,T622D)
 26N P     2 F
 27N A   600 D  first address
 28N π       F
 29N U   800 D  (†U702D,U642D,U622D)
 30N P   200 F
 31N O       F
 32N T  1000 D  (T†902D,T842D,T822D)
 33N A   600 D
      T   100 K
```

division subroutine

```
      T   200 K   subroutine called 2 times
      G       K
    0 A     3 F   Mparameter U2F plant link
    1 T    17 θ   plant link
    2 A    24 N   T600D, clear 600D...798D (ss)
10→ 3 U     7 θ
    4 S    25 N   T800D
    5 E    11 θ   exit
    6 T       F
    7 (T   600 D) clear work space
    8 A     7 θ
    9 A    26 N   P2F
   10 E     3 θ   end of clear loop
 5→11 T       F
   12 A      πN   80=4×4×5
   13 T   600 D   first numerator
   14 A   12πN   long 1
   15 U   18πN
   16 S    4πN   1435 i=1,3,5,...,1435
   17 (Z      F) end of subroutine
   18 T       F
   19 T   20πN   ws
   20 A    2πN   25
   21 T    4πH   set divisor
```

	22	A	27	N	A600D division loop
48→	23	U	31	θ	→A600D loop back point
	24	A	28	N	U−A=7−28=39−28=11=π πF
	25	U	38	θ	→U600D
	26	S	29	N	U800D last address
	27	E	49	θ	exit from division loop
	28	T		F	
	29	A	20	πN	ws
	30	T		πH	upper dividend
	31	(A	600	D)	by 23
	32	T	2	πH	lower dividend
	33	A	33	θ	
	34	G	100	F	division subroutine
	35	A		πH	remainder
	36	T	20	πN	ws
	37	A	2	πH	quotient
	38	(U	600	D)	
	39	S	12	πN	long 1, zero test
	40	E	45	θ	no overflow
	41	T		F	
	42	A	31	θ	
	43	A	26	N	P2F
	44	T	27	N	update next start address
40→	45	T		F	come here from no overflow
	46	A	31	θ	advance address
	47	A	26	N	P2F
	48	G	23	θ	loop back
27→	49	T	20	πN	ws, 2nd loop acc = 0
	50	A	18	πN	i
	51	T	4	πH	divisor
	52	A	27	N	A600D first address
87→	53	U	64	θ	A600D loop back comes here
	54	A	30	N	P200F
	55	U	70	θ	A800D
	56	A	31	N	T−A = 37−28=9=O
	57	U	74	θ	T800D
	58	U	84	θ	T800D
	59	S	32	N	T1000D end check
	60	E	88	θ	return, goto adveace address
	61	T		F	
	62	A	20	πN	ws
	63	T		πH	
	64	(A	600	D)	
	65	T	2	πH	
	66	A	66	θ	
	67	G	100	F	division subroutine
	68	A		πH	remainder
	69	T	20	πN	ws
	70	(A	800	D)	add to ts
	71	(A	2	πH)	A2πH↔S2πH switch
	72	E	84	θ	no overflow
83→	73	A	16	πN	2^{34}, clear overflow bit
	74	(T	800	D)	carry assimilation
	75	A	74	θ	
	76	S	26	N	P2F
	77	U	74	θ	
	78	U	84	θ	
	79	S	31	N	T−A
	80	T	81	θ	
	81	(A	800	D)	
	82	(A	12	πN)	long 1 A12πN↔S12πN switch
	83	G	73	θ	loop back to add 2^{34}
72→	84	(T	800	D)	
	85	A	64	θ	advance address
	86	A	26	N	P2F
	87	G	53	θ	
60→	88	T		F	
	89	A	22	N	reverse binary switch
	90	S	71	θ	
	91	T	71	θ	

```
 92 | A   23  N |  reverse binary switch
 93 | S   82  θ |
 94 | T   82  θ |
 95 | A   18 πN |  i
 96 | A   14 πN |  long 2
 97 | E   15  θ |
    T  300  K     main routine
    G      K
  0 | A   22  N |  set binary switch
  1 | A   71  M |  A2πH+S2πH
  2 | T   22  N |
  3 | A   23  N |  A12πN+S12πN
  4 | A   82  M |
  5 | T   23  N |
12→ 6 | T      F |
  7 | (T 800 D) |  clear 800D...998D (ts)
  8 | A    7  θ |
  9 | A   26  N |  P2F
 10 | U    7  θ |
 11 | S   32  N |  T1000D
 12 | G    6  θ |
 13 | T      D
 14 | A   14  θ |
 15 | G      M |  1st subroutine call
 16 | A    6 πN |  80←956
 17 | T     πN |
 18 | A    8 πN |  25←57121
 19 | T    2 πN |
 20 | A   10 πN |  1435←425
 21 | T    4 πN |
 22 | A   33  N |
 23 | T   27  N |  reset start address
 24 | A   24  θ |
 25 | G      M |  2nd subroutine call
 27 | H    8  F |
 28 | T      F |
 29 | E   34  F |  return to initial orders
    E300KPF       start program
```

long fraction print routine

$\pi = 16\tan^{-1}(1/5) - 4\tan^{-1}(1/239);$

$ts \leftarrow 0;$
$ss \leftarrow 16 \times 5;$
$f(5, 1433, +);$
$ss \leftarrow 4 \times 239;$
$f(239, 433, -);$
$ts;$
where $f(z, k, \odot)\{$
  for$(i \leftarrow 1; i < k; i \leftarrow i + 2)\{$
    $ss \leftarrow ss/z^2;$
    $ts \leftarrow ts \odot (ss/i);$
    $\odot \leftarrow \odot = +?- : +\}\}$

Long fraction print

Prints out a long fraction stored in 802D, 802D, ...,998D. A single integral digit is stored in 800D. For example, the value of *pi1000* is printed as:

```
3.14159265358979323846264338327950288419716939937510
  58209749445923078164062862089986280348253421170679
  ...
  18577805321712268066130019278766111959092164201989
```

```
      PKT45K          interlace print 802...998
   45 ‖ P    56  F ‖   set H N parameters
   46 ‖ P    64  F ‖
      T56πZPF         clears sandwich digit
      T    56  K
  0H ‖ P3584FP46462D   −10^10, later used for carry
  2H ‖ (PFPF     )‖    used for number to be printed
      T    64  K      long multiplication routine
      G        K
    0 ‖ A    23  θ ‖   A+V=G
    1 ‖ T     7  θ ‖   plant link
    2 ‖ A    24  θ ‖   last word address
22→ 3 ‖ U    13  θ ‖
    4 ‖ A    25  θ ‖   V+Y=T
    5 ‖ U    19  θ ‖
    6 ‖ S    26  θ ‖   first word address
    7 ‖ (Z       F)‖   test end and return
    8 ‖ T        D ‖   clear acc
    9 ‖ A       πH ‖   carry
   10 ‖ R        F ‖   right shift 34 bits
   11 ‖ R        F ‖
   12 ‖ R     4  F ‖
   13 ‖ (V       D)‖   multiply words V1000D V998D ...
   14 ‖ U       πH ‖   store carry
   15 ‖ S       πH ‖   clear acc
   16 ‖ L        F ‖   left shift 34 bits
   17 ‖ L        F ‖
   18 ‖ L    64  F ‖
   19 ‖ (T       D)‖   store words T1000D T998D ...
   20 ‖ A    13  θ ‖
   21 ‖ S    27  θ ‖   decrease address
   22 ‖ G     3  θ ‖
   23 ‖ V     2  F ‖   constant
   24 ‖ (V 1000  D)‖   last word
   25 ‖ Y        F ‖
   26 ‖ (T  802  D)‖   first word
   27 ‖ P     2  F ‖
      T100πZPF         clears sandwich digit
      T   100  K
      G        K
    0 ‖ P 39680 F ‖   −10^9
    1 ‖ P 63628 D ‖
    2 ‖ P    20 F ‖   (†P10F,P4F,P2F)
    3 ‖ P     5 F ‖
    4 ‖ (π       F)‖   figs, 20 counter
    5 ‖ (M       F)‖   ., 5 counter
    6 ‖ θ        F ‖   cr
    7 ‖ Δ        F ‖   lf
    8 ‖ φ        F ‖   sp
    9 ‖ Q        F ‖   1 code unit
   10 ‖ V 1000   D ‖   (†V900D,V840D,V820D)
   11 ‖ T  802   D ‖
   12 ‖ X   18   F ‖   (†X8F,X2F,XF)
   13 ‖ S       πH ‖   prgram starts here
   14 ‖ T        D ‖
   15 ‖ H        D ‖   10^10 to multiplier register
   16 ‖ A  800   F ‖   first digit in 800D
   17 ‖ L 1024   F ‖
   18 ‖ T        F ‖   first digit code to 0F
   19 ‖ O    4  θ ‖   figs
```

```
      20   O     6  θ   cr
      21   O     7  θ   lf
      22   O        F   print the first digit 2 or 3
      23   O     5  θ   print .
      24   A    24  θ
      25   G    64  F   first multiplication
      26   T        F
      27   S     2  θ   20
72 →  28   T     4  θ   20 counter
      29   S     3  θ   5
      30   T     5  θ   5 counter
      31   A     πH
      32   T    2πH
      33   A    10  θ   V1000D
60 →  34   U    24  N   first word
      35   S    12  θ   X18F
      36   U    26  N
      37   S    11  θ
      38   G    61  θ
      39   T        F
      40   A    40  θ
      41   G    64  F   second multiplication
      42   T        F
50 →  43   T        F   digit code
      44   A    2πH
      45   A     πθ   −10^9
      46   G    51  θ
      47   T    2πH
      48   A        F   digit code
      49   A     9  θ   + code unit
      50   E    43  θ
46 →  51   S     πθ   ┐ multiply by 10
      52   U    2πH  │
      53   L     1  F  │
      54   A    2πH  │
      55   L        D  ┘
      56   T    2πH
      57   O        F   print a digit code
      58   A    24  N   V1000D
      59   S     2  θ   P20F
      60   G    34  θ   loop back
38 →  61   T        F
      62   A     5  θ   5 counter
      63   A     2  F   +1
      64   G    30  θ
      65   O     6  θ   cr
      66   O     7  θ   lf
      67   A     4  θ   20 counter
      68   A     2  F   +1
      69   E    73  θ
      70   O     8  θ   sp
      71   O     8  θ   sp
      72   G    28  θ
69 →  73   Z        F   stop
           E13ZPF
```