```
start,image<
slip<
[Calculation of PI and E

Based on DEMON-5.

mk at lemo.dk March 2015.

The program stops after reading.

You can change the following before starting
the program by pressing Normal Start:

Raddr: First drum track, default 103
Rtæl:  No. of tracks per register, default 4
KA:    0: Calculate PI, 1: Calculate E

Maximum capacity is obtained by enable writing on tracks
0, 1-31 and setting Raddr to zero. The maximum
for Rtæl is 64 for PI and 160 for E, corresponding to
25590 or 63990 decimals, respectively.

The primitive operations can be followed on the
indicator during execution:

OA:  Divide
OB:  Copy
TA:  Add
TB:  Subtract

Timing for PI:

Tracks  sec
 1        86
 2       247
 3       486
 4       802
 5      1192
 6      1661
 7      2206
 8      2825
 9      3522
10      4296
11      5145
12      6070
13      7072
64    159279 = 44h14m

Disk:

Tracks  sec
 4       819
 8      3022
64    623012 = 173h03m

E:

Tracks  sec
 1        36
 2        95
 3       180
 4       288
 5       419
 6       572
 7       746
 8       943
 9      1160
```

```
10     1398
11     1657
12     1936
13     2235
14     2554
15     2893
16     3253
17     3631
160  236213 = 65h37m
```

After reading the program by SLIP, press HP and punch the program as tape readable by r<:

```
outparam,binin,image<
binout,n<
outparam,exit,10<

]

b a50,b50,c50,d50,e50
e1=4     ; step: No. of tracks per register
e2=1     ; Use timing on simulator
b1=800   ; integer array A1[0:39]
b2=840   ; integer array A2[0:39]
b3=880   ; integer array B1[0:39]
b4=920   ; integer array B2[0:39]
i=10
b a20                      ; Main block
d8:  hs  1
     hv  d9                ; x,list,free<
     tx;
     tlist;
     tfree;
     qqf,
d9:  tk  30    , gr  c30   ; where(‹<free›, FREE);
d0:  pmn c9    X           ; begin main program; R:=step; M:=0;
     ck  20    , ar  c30   ; Raddr:=FREE; Rtæl:=step;
     zq                    ; stop;
     cl  10    , ck  -30   ;
     gr  c9    , cl  -10   ;
     gr  c30               ;
     vy  17                ; select(17);
<e2
     zl                    ; clock count:=0;
>    pi  0                 ; IN:=0;
     hv  a19           LKA ; if -, KA then begin comment PI TO;
     hs  d1                ; ASSIGN(0, asize, 0);
     qq  0.39
     qq  c3
     qq  0.39
     hs  d1                ; ASSIGN(48, t1size, 1);
     qq  48.39
     qq  c4
     qq  1.39
     hs  d3                ; DIVIDE(t1size, 1, 18, out1)
     qq  c4
     qq  1.39
     qq  18.39
     qq  c10
     hs  d1                ; ASSIGN(32, t2size, 2);
     qq  32.39
     qq  c5
     qq  2.39
     hs  d3                ; DIVIDE(t2size, 2, 57, out2);
     qq  c5
     qq  2.39
     qq  57.39
```

```
     qq  c11
     hs  d1                ; ASSIGN(20, t3size, 3);
     qq  20.39
     qq  c6
     qq  3.39
     hs  d3                ; DIVIDE(t3size, 3, 239, out3);
     qq  c6
     qq  3.39
     qq  239.39
     qq  c12
     arn c1    , gr a1
     gr  a2                ; m := factor := 1;
a10: hs  d4                ; COPY(t1size, 1, ssize, 4)
     qq  c4
     qq  1.39
     qq  c7
     qq  4.39
     arn c11
     hv  a11           NZ  ; if -,out2 then
     hs  d5                ; ADD(t2size, 2, ssize, 4);
     qq  c5
     qq  2.39
     qq  c7
     qq  4.39
a11: arn c12
     hv  a12           NZ  ; if -,out3 then
     hs  d6                ; SUB(t3size, 3, ssize, 4);
     qq  c6
     qq  3.39
     qq  c7
     qq  4.39
a12: arn a1    , gr  a13
     hs  d3                ; DIVIDE(ssize, 3, m, out);
     qq  c7
     qq  4.39
a13: qq  [m]
     qq  c13
     arn a2
     hv  a14           LT  ; if factor ≥ 0 then
     hs  d5                ; ADD(ssize, 4, asize, 0);
     qq  c7
     qq  4.39
     qq  c3
     qq  0.39
     hv  a15               ; else
a14: hs  d6                ; SUB(ssize, 4, asize, 0);
     qq  c7
     qq  4.39
     qq  c3
     qq  0.39
a15: arn a2
     mt  -1 D
     gr  a2                ; factor := - factor
;
;    sy  64
;    hs  d2                ; PRINT(asize, 0);
;    qq  c3
;    qq  0.39

     hs  d3                ; DIVIDE(t1size, 1, 324, out1);
     qq  c4
     qq  1.39
     qq  324.39
     qq  c10
     arn c11
     hv  a16           NZ  ; if -,out2 then
     hs  d3                ; DIVIDE(t2size, 2, 3249, out2);
```

```
     qq  c5
     qq  2.39
     qq  3249.39
     qq  c11
a16: arn c12
     hv  a17           NZ  ; if -,out3 then
     hs  d3                ; DIVIDE(t3size, 3, 57121, out3);
     qq  c6
     qq  3.39
     qq  57121.39
     qq  c12
a17: arn a1    , ar  a3
     gr  a1                ; m:=m+2;
     arn c10
     hv  a10           LZ  ; if -,out1 then goto L1;
     hv  a18               ; end PI TOelse begin
a19: hs  d1                ; ASSIGN(2, asize, 0);
     qq  2.39
     qq  c3
     qq  0.39
     hs  d1                ; ASSIGN(1, t1size, 1);
     qq  1.39
     qq  c4
     qq  1.39
     arn c1    , gr a20    ; m := 1;
a21: arn a20   , ar c1     ;L1: m := m+1;
     gr  a20
     hs  d3                ; DIVIDE(t1size, 1, m, out);
     qq  c4
     qq  1.39
a20: qq  [m]
     qq  c13
     hs  d5                ; ADD(t1size, 1, asize, 0);
     qq  c4
     qq  1.39
     qq  c3
     qq  0.39
     arn c13
     hv  a21           LZ  ; if -,out then goto L1; end E TO;
<e2
a18: hs  d7                ; PRINT CLOCK;
     sy  64
×
a18: sy  64
>    hs  d2                ; PRINT(asize, 0);
     qq  c3
     qq  0.39
     sy  64

     vk  0                 ; wait drum
     hv  d0                ; restart program
a1:  qq                    ; m
a2:  qq                    ; factor
a3:  qq 2.39
e                         ; Main block

c1:  qq  1.39
c2:  qq  10000000000.39
c8:  qq  40.39
c9:  qq  e1.39             ; step
c17: qq 1000000000.39      ; MODUL:10
c19: qq 10.39              ; 10
c21: qq 6.39               ; 6 groups per line
c24: qq  b1                ; A1
c25: qq  b2                ; A2
c28: qq  b3                ; B1
c29: qq  b4                ; B2
```

```
c30: qq  0                  ; FREE
c3=700    [asize]
c4=701    [t1size]
c5=702    [t2size]
c6=703    [t3size]
c7=704    [ssize]
c10=705   [out1]
c11=706   [out2]
c12=707   [out3]
c13=708   [out]
c14=709   [group]
c15=710   [count]
c16=711   [n]
c18=712   [DIVISOR]
c20=713   [asize-1]
c22=714   [carry]
c23=715   [nt]
c26=716   [c2]
c27=717   [c3]
[
       procedure ASSIGN(n, asize, na);

     hs  d1                 ; Call ASSIGN
     qq  n.39               ; [s1|Value of n
     qq  <addr. of asize>   ; [s2|Address of asize
     qq  na.39              ; [s3|Value of na
                            ; Return to s+4
]
b a10
d1:  pp  b1    , vk  0      ; waitfordrum;
;    sy  64    , sy  53     ; sy(e)qual;
     pa  a2    t 39         ; for i:=0 step 1 until 39 do
a1:  grn p     , pp  p1     ; A[i]:=0;
a2:  bt  39    t -1         ;
     hv  a1
     pm  s3
     mln c9    X            ; R:=na×step
     ck  30
     ar  c30                ; Raddr:=na×step+FREE
     sr  1     D            ; Raddr:=na×step+FREE-1
     ga  a3                 ; nt:=Raddr;
     arn c9    , sr  c1     ;
     ck  30    , ga  a4
a3:  vk  0     t 1         ; for i:=step-1 step -1 until 0 do
     sk  b1                 ; begin put(A, FREE, nt); nt:=nt+1;
     bs  (a4)  t 0          ; if i≤0 then
     hv  a4                 ;
     arn s1    , gr 39b1    ; A[39]:=n;
a4:  bt  [step] t -1
     hv  a3                 ; end for i;
     xrn       , arn s1     ; M:=0; R:=n;
     hv  a5            LZ   ; asize:=if n=0 then 0
     pm  c8    , mln c9     ; else step×40;
a5:  gm (s2)
     hr  s4
e                           ; end ASSIGN;
[
       procedure PRINT(asize, na);

     hs  d2                 ; Call PRINT
     qq  <addr. of asize>   ; [s1|Address of asize
     qq  na.39              ; [s2|Value of na
                            ; Return to s+3

                            ; Indicator:
                            ; PA: first
]
```

```
b a20
d2:  pi  32    t 991        ; first := true;
;    sy  64    , sy  39     ; sy(p)rint;
     grn c14                ; group:=0;
     pa  a10   t 39         ; c:=39;
     pm  c8                 ;
     mln c9    X            ; R:=step×40;
     sr  c1    , gr  c15    ; count:=step×40-1;
     gr  c20                 ; asize-1:=count;
     arn s2    , ar  c1     ; R:=na+1;
     xr                     ;
     mln c9    X            ; R:=(na+1)×step
     ck  30
     ar  c30                ; Raddr:=na×step+FREE
     ga  a11                ; ta:=Raddr;
a11: vk  0     t -1         ;
     lk  b1     , vk (a11)  ; get(A, FREE, ta);
a12: bs (a10)   t -1        ; if c≤-1 then
     hv  a13
     pa  a10    t 39        ; c:=39;
     vk (a11)   t -1        ; ta:=ta-1;
     lk  b1     , vk (a11)  ; get(A, FREE, ta);
a13: is (a10)   , arn sb1   ; R:=A[c];
     gr  c16    , arn c17    ; n:=R;
     gr  c18     , qqn       ; DIVISOR:=MODUL:10; R:=0;
     ar  16     D      NPA  ; if -,PA then Raddr:=16;
     ga  a16                ; space := if first then 0 else 16;
     pa  a14    t 9         ; for i:=9 step -1 until 0 do
a15: pm  c16    , dln c18    ; M:=n mod DIVISOR; R:=n : DIVISOR;
     gm  c16                ; n:=n mod DIVISOR;
     hv  a16           LZ   ; if digit ≠ 0 then
     ck  -10    , ga  r1    ;
     sy  0                 ; writechar(digit);
     pi  0      t 991       ; first:=false;
     pa  a16    Vt 16       ; space:=16;
a16: sy  0                 ; else writechar(space);
     arn a14    , ca 5      ; if i=5 then
     sy  0                  ; writechar(0);
     pm  c18     , dln c19    ; R:=DIVISOR:10;
     gr  c18                 ; DIVISOR:=R;
a14: bt  0      t-1         ; end for i;
     hv  a15
     arn c15    , sr  c20    ; R:=count-(asize-1);
     hv  a17           NZ   ; if count=asize-1 then begin
     sy  59                 ; writetext(⟨<.⟩);
     pi  0      Vt 991       ; first := false;
a17: sy  0                  ; end else writechar(0);
     arn c14    , ar  c1    ; R:=group+1;
     gr  c14    X           ; group:=R; M:=group;
     dln c21     X           ; R:=group mod 6;
     sy  64            LZ   ; if R=0 then writecr;
a10: qq  [c]    t -1        ; c:=c-1;
     arn c15    , sr  c1    ; count:=count-1;
     gr  c15
     hv  a12           NT   ; end for count;
     hr  s3

e                           ; end PRINT;
[
        procedure DIVIDE(asize, na, n, empty);

     hs  d3                 ; Call DIVIDE
     qq  <addr. of asize>   ; [s1|Address of asize
     qq  na.39              ; [s2|Value of na
     qq  n.39               ; [s3|Value of n
     qq  <addr. of empty>   ; [s4|Address of empty
                            ; Return to s+5
```

```
                              ; Indicator:
                              ; PA: first
                              ; PB: short (c<20)
]
b a20
d3:  pi  544                  ; first := true; short := false;
;    sy  64    , sy  52       ; sy(d)ivide;
     grn c22                   ; carry := 0;
     arn (s1)  , sr  c1       ; R:=asize-1;
     xr        , dln c8       ;
     ar  c1    , gr  c23       ; nt:=1+(asize-1):40;
     xr        , ck  30       ; Raddr:=(asize-1)mod 40;
     ga  a10                  ; c:=Raddr;
     bs  (a10)  t 19          ; if c≤19 then
     hv  r2
     pi  16    t 1007         ; short := true;
     pm  s2
     mln c9    X              ; R:=na×step
     ar  c23    , ck  30       ; Raddr:=na×step+nt
     ar  c30                  ; Raddr:=na×step+FREE
     ga  a11                  ; nt:=na×step+FREE+1
a11: vk  0     t -1           ; nt:=nt-1; vk(nt);
     lk  (c25)  , vk  (a11)    ; get(A, FREE, nt);
a13: arn c24    , pm  c25       ; Swap buffers
     gr  c25    , gm  c24       ;
     arn c24    , ck -10
     gt  a12   , gt  a14      ; insert address of buffer
     arn a11   , ga  a15      ; Store track for write
a12: is  (a10)  , arn sb1      ; R:=A[c];
     pm  c22    , ml  c2       ; RM:=(cell+carry×MODUL);
     dl  s3     , gm  c22      ; carry:=rem.
a14: is  (a10)  , gr  sb1      ; A[c]:=quotient;
     hv  a16         NPA      ; if first then
     pi  0    Vt 991   NZ     ; if cell>0 then first:=false
     arn c1    , sc  (s1)     ; else asize:=asize-1;
a16: arn c23    , sr  c1       ; if (c=20 ∨ short) ∧ count≠1then
     hv  a10            LZ    ;
     hv  a17            LPB   ;
     arn a10   , nc  20       ;
     hv  a10                  ; begin
a17: vk  (a11)   t -1         ; nt:=nt-1; vk(nt);
     lk  (c25)                 ; get(B,FREE,nt);
     pi  0       t 1007       ; short:=false end;
a10: bt  [c]     t -1         ;
     hv  a12
a15: vk  0     , sk  (c24)    ;
     pa  a10   t 39           ; c:=39;
     arn c23    , sr  c1       ; R:=count-1;
     gr  c23                   ; count:=R;
     hv  a13            NZ    ;
     pm  c1                   ; M:=1;
     grn (s4)  , arn (s1)     ; empty:=false; R:=asize;
     gm  (s4)           LZ    ; if asize=0 then empty:=true;
     hr  s5

e                             ; end DIVIDE;
[
        procedure COPY(asize, na, bsize, nb);

     hs  d4                   ; Call COPY
     qq  <addr. of asize>     ; [s1|Address of asize
     qq  na.39                ; [s2|Value of na
     qq  <addr. of bsize>     ; [s3|Address of bsize
     qq  nb.39                ; [s4|Value of nb
                              ; Return to s+5
```

```
]
b a20
d4:  pm  s2    , pi  256
;    sy  64    , sy  51    ; sy(c)opy;
     mln c9    X           ; R:=na×step
     ck  30
     ar  c30
     sr  1     D           ; Raddr:=na×step+FREE
     ga  a1                ; t1:=Raddr;
     pm  s4
     mln c9    X           ; R:=nb×step
     ck  30
     ar  c30
     sr  1     D           ; Raddr:=na×step+FREE
     ga  a2                ; t1:=Raddr;
     arn c9    ,  ck  30   ; Raddr:=step-1
     sr  1     D
     ga  a3
a1:  vk  [t1]  t 1         ; t1:=t1+1; vk(t1);
     lk  b1                ; get(A, FREE, t1);
a2:  vk  [t2]  t 1         ; t2:=t2+1; vk(t2);
     sk  b1                ; put(A, FREE, t2);
a3:  bt  [count]t -1       ;
     hv  a1
     arn (s1)   , gr  (s3) ; bsize:=asize;
     hr  s5
e                          ; end COPY;
[
        procedure ADD(bsize, nb, asize, na);

     hs  d5                ; Call ADD
     qq  <addr. of bsize>  ; [s1|Address of bsize
     qq  nb.39             ; [s2|Value of nb
     qq  <addr. of asize>  ; [s3|Address of asize
     qq  na.39             ; [s4|Value of na
                           ; Return to s+5

]
b a20
d6:  pi  64    , arn a22    ; subtract: TB:=1;
     ab  a18   , hh  a24
d5:  pi  128   , arn a23   ; add: TA:=1;
a24: mb  a18   , gr  a18
     arn (s3)  , sr (s1)   ; if bsize>asize then asize:=bsize;
     hv  r2            NT  ;
     arn (s1)  , gr (s3)   ;
     arn (s3)  , sr  c1    ; R:=asize-1;
;    sy  64    , sy  49    ; sy(a)dd;
     xr        , dln c8    ;
     ar  c1    , gr  c23    ; nt:=1+(asize-1):40;
     grn c22    , arn c1    ; carry := 0;
     gr  c15                ; count := 1;
     pt  a10   t 39        ; nc:=39;
     arn (s3)  , gr  c26    ; c2:=asize;
     grn  c27               ; c3:=0;
     grn (s3)              ; asize:=0;
     pm  s2
     mln c9    X           ; R:=nb×step
     ck  30
     ar  c30
     sr  1     D           ; Raddr:=nb×step+FREE
     ga  a11               ; t2:=Raddr;
     pm  s4
     mln c9    X           ; R:=na×step
     ck  30
     ar  c30
     sr  1     D           ; Raddr:=na×step+FREE
```

```
     ga  a12                ; t1:=Raddr;
a11: vk  [t2]  t 1          ; t2:=t2+1; vk(t2)
     lk (c29)                ; get(B, FREE, t2);
a12: vk  [t1]  t 1          ; t1:=t1+1; vk(t1);
     lk (c25)   , vk (a12)  ; get(A, FREE, t1); waitfordrum;
a17: pp  0                  ;L1: c1:=0;
     arn c24    , pm  c25    ; swap
     gr  c25    , gm  c24
     arn c28    , pm  c29
     gr  c29    , gm  c28
     arn c24    , ga  a13  ; Store addresses of buffers
     arn c28    , ga  a18
     arn a12   , ga  a15   ; Store track for write
a13: arn pb1   , ar  c22     ;L2: R:=A[c1]±B[c1]+carry;
a18: ar  pb2  , gr  (a13)    ; A[c1]:=R;
     hv  a27           NT  ; if R<0then begin
     srn c1    , gr  c22    ; carry:=-1;
a25: arn c2    , ac  (a13)   ; A[c1]:=A[c1]+MODUL;
     hv  a26
a27: sr  c2                 ; R:=R-MODUL;
     hv  a14           LT  ; if R≥0 then begin
     arn c1    , gr  c22    ; carry:=1;
a19: arn c2    , sc  (a13)   ; A[c1]:=A[c1]-MODUL;
     hv  r2                 ; end else
a14: grn c22                 ; carry:=0;
a26: arn c1    , ar  c27    ; c3:=c3+1;
     gr  c27    X            ; M:=c3
a20: arn (a13)                ; R:=A[c1];
     gm (s3)           NZ  ; if R≠0 then asize:=M;
     pp  p1                 ; c1:=c1+1;
     arn c15    , sr  c23    ; if count≠nt then
     hv  a10           LZ  ; begin
     arn p     D            ; Raddr:=p
     nc  13    , hv  a21   ; if p=13 then
     vk (a11)  t 1
     lk (c29)   , hv  a10
a21: nc  26    , hv  a10
     vk (a12)  t 1
     lk (c25)
a10: bs  p     t  [nc]      ; if c1≤nc then goto L2;
     qq  V
     hv  a13
     arn c22
     hv  a15           LZ  ; end
     bs  p     t 39         ; if p≤39 ∧ carry ≠ 0 then goto L2;
     qq  V
     hv  a13
a15: vk  0     , sk (c24)    ; put(A, FREE, t1);
     arn c15    , ar  c1     ; count:=count+1;
     gr  c15    , sr  c23    ; if count=nt then
     hv  a16           NZ  ;
     arn c26    , sr  c1     ; R:=c2-1
     xr        , dln c8     ;
     xr        , ck  20     ; Raddr := (c2-1) mod 40;
     gt  a10                ; nc := Rtæl;
a16: arn c22
     hv  a17           NZ  ; if carry>0 then goto L1;
     arn c23    , sr  c15    ; R:=nt-count;
     hv  a17           NT  ; if R≥0 then goto L1;
     hr  s5
a22: qq  16384.39              ; ar -> sr
a23: qq  -16385.39            ; sr -> ar
e                           ; end ADD;
<e2
[
        procedure PRINT CLOCK;
```

```
]
b a20
d7:  zl        , tkf -29   ; R:=clock in seconds;
     pi  32    t 991       ; first := true;
     sy  64                ; writecr;
     gr  c16     , arn c17   ; n:=R;
     gr  c18     , qqn      ; DIVISOR:=MODUL:10; R:=0;
     ar  16     D      NPA ; if -,PA then Raddr:=16;
     ga  a16               ; space := if first then 0 else 16;
     pa  a14    t 9        ; for i:=9 step -1 until 0 do
a15: pm  c16     , dln c18   ; M:=n mod DIVISOR; R:=n : DIVISOR;
     gm  c16                ; n:=n mod DIVISOR;
     hv  a16           LZ  ; if digit ≠ 0 then
     ck  -10    , ga  r1   ;
     sy  0                ; writechar(digit);
     pi  0      t 991      ; first:=false;
     pa  a16    Vt 16      ; space:=16;
a16: sy  0                ; else writechar(space);
     pm  c18     , dln c19   ; R:=DIVISOR:10;
     gr  c18                ; DIVISOR:=R;
a14: bt  0      t-1        ; end for i;
     hv  a15
     sy  0      , sy  18   ; writetext( sec);
     sy  53     , sy  51   ;
     hr  s1
e
>
e d8                       ; end program;
```