Addertest

i test 1 skrives 1 i pos 19 og 39
i alle celler (LI)

i test 2 nulstilles först R reg. og
ved hvert gennemlöb adderes nu
et i henholdsvis pos 19 og 39.
Ved hvert stop (manuelt) skal de
to halvdele af R være eens (indtil
overlöb fra sidste halvdel når förste
halvdel)

OVERSIGT OVER BESKREVNE TESTPROGRAMMER.

B1: (Topsøe's tromletest). Tester med tilfældige tal kanal 1-319. Mærkebits testes ikke.

B2: Trykning af et smukt bitmønster. Udgaar i nærmeste fremtid.

B3: Trykning af Kanal 0 som bitmønster. Udgaar i nærmeste fremtid.

B4: Ferritlagertest med manuelt indsat bitmønster. Celle 60-1022 testes, men ikke mærkebits.

B5: Test af kanal 0. Kanal 0 sammenlignes med en korrekt version, der indlæses som en del af programmet.

B6: Ferritlagertest, der ogsaa tester mærkebits. Der forsøges skrivning af ettere og nuller i cellerne 50-1023 og 0-40.

B7: (Topsøe's flydende test). Test af operationerne arf, srf, mkf og dkf.

B8: Undersøgelse af tromletransporttiden. Udgaar snarest.

B9: Skrivemaskinetest. Et indtastet tegn gentages ni gange. Tester ogsaa caseskift.

B10: Ferritlagertest med mærkebits. Bruger enten automatisk genererede bitmønstre eller manuelt indsatte. Tester celle 512-1023.

B11: Som test B10, men test af celle 0-511.

B12: Indikatortest. Tester indikatoroperationerne og saa vidt vides alle indikatorbetingelserne.

B13: (Magnus tromletest). Tester tromlen med tilfældige tal og kan foretage simultanregning. Mærkebits testes.

B14: Test af læser. Tester læsning af samtlige gyldige hulkombinationer.

B15: Test af hulning og læsning af 5-kanals tape i TELEX kode.

16 udgår

B17: Printertest. Trykker samtlige tegn i alle kolonner.

B18: Disktest. Test af diskfile som tromle. Tester et defineret omraade af disken med adressen paa det paagældende ord.

B25: Disktest. Test af diskfile som tromle. Tester hele disken med enten manuelt indsat information eller med spornummer som information.

A200: Program til hulning af strimler til indlæsning med kanal 0.

A201: Perforatortest. Huller en strimmel, der testes ved umiddelbar læsning.

A203: (Buffertest 2). Tester bufferen med tilfældige tal. Tester ogsaa mærkebits.

A204: (Buffertest 3). Tester bufferen efter samme principper som B10 og B11.

A205: Sorteringstest. Tester sortering paa 3 baandstationer.

A206: Program til hulning af strimmel til B14.

A208: Bundgaards rutinetest af GIER.

A209: Test af adressering paa diskfile tilsluttet bufferen.

A210: Test af diskfile tilsluttet buffer. Der testes med manuelt indsat information.

A211: Hurtig sumcheck af diskfile tilsluttet buffer.

A212: Test af læsning fra diskfile tilsluttet som tromle. Læser med størst mulig bevægelse af armene.

A213: Test af hovedvalg paa diskfile tilsluttet buffer.

A215: Test af diskfile paa buffer. Test af simultanregning. Valg af spor i tilfældig orden ved læsning. Hele statusordet undersøges.

Februar 1967, J.F.M.

KANAL 0 i HJÆLP 3

```
 93   8.8.69  849
slip<
vk0
lk100
sy1
zq
e10
1
 94   8.8.69  e14
print.p100..139<

100. qq               .~~hh   23~~
101. it   960         .pt    1
102. qq               .gk    2
103. vy   529         .hv    6
104. ann  s64         V   MK  t479
105. ml  (p64)        VD LPB t921
106. gi   3               IPC
107. gm   6               MPC
108. gs   7               IOB
109. gr   8               MOB
110. ga   9           .tl   -39
111. pm   s               IRB
112. gr   11          .hr    r1 [1̸13]
113. gp   12          .arn   4
114. sr   -1          .tk    21
115. hv   29              NZ
116. hv   33              NA
117. arn  5           .pm   -1
118. ga   19          .tk    10
119. sy   -1          .ck   -4
120. nc               .hv    18
121. lyn  -1          .gm   -1
122. bs   1           .hv    27
123. gs   10          .vyn   528
124. tl   -6          .ca
125. ly   r4 [1̸29]    .hs    r-1 [1̸24]
126. gm   s3               M   t-1
127. ca               .hv    33
128. nc   17          .hv    17
129. vk   960         .it    1
130. vk   294         .it    40
131. sk               .it   -1
132. ncn  24          .hh    29
133. vkf  960         .vkf   1
134. lk   40          .vkn   38
135. ar   37          .pa    22
136. ar   2           D  LA
137. hv  (35)         D  NB  t1
138. hv  (17)         D  NZ  t-1
139. arn  1               IQC
```

start primitiv læseprogram → 123

61	51
18	25
59	50

indlæsning af „C„TESTS indsæt celle 23 - 24 - 25

BY = 0

Start celle 23

GIER SYSTEM LIBRARY
1.2 Hardware Tests
A/S Regnecentralen
TA. 14/10-64.

Order No. 24

Udskrift af kanal 0 i HJÆLP-systemet.

```
[   0] qq
[   1] it    1      ,pa    10      gem h-bit
[   2] gk    1      ,vy    r-497   bit 20-39 er konstant for spærreord
                                     gem tk og by, spær hp-knap
                                     vælg skrivemaskine in/output
[   3] gi    2         IPC         gem in, indicer A og B
[   4] gm    3         MPC         gem M, A og B
[   5] vk    32        IOB t-1     vk er irrelevant, indicer overløb
[   6] gr    4         MOB         gem R og R[00]
[   7] tl    -39    ,pm    24      2R[00]+R[0]->R[39],M[0-19]:=-1
[   8] gr    7      ,tln   -19     gem R[00],R[0],2⅄(-19)-2⅄(-39)->M
[   9] arn   2      ,cm    -1      r:=spærreord, sammenlign operationsdel
                                     i celle -1 med spærreord
[  10] pmfn  r         IRB         R:=0, RB:=h, M irrelevant
[  11] pi    66        IZA t-511   in=-,sp,0,0,1,0,0,0,0,1,h
[  12] hh    17        NZA         hop hvis hp-knap spærret
[  13] gp    8      ,pp    38      gem p, p:=38
[  14] vk    319          t-1      )
[  15] sk    -24          t-40     ) gem celle 40-959 paa kanal 295-318
[  16] bt    23           t-1      )
[  17] hv    14     ,pp    294     hvis hp-knap spærret then p:=294
[  18] gs    6      ,gin   9       gem s, gem in, R:=0
[  19] vk    25     ,lk    -64     læs kanal 25
[  20] vk    p      ,lk    -73     læs kanal med oprindeligt indhold
                                     af celle 0-39
[  21] vk   (5)     ,lk    -124    )
[  22] ps    41     ,ud    5       )
[  23] ps    s-1    ,ar    s-125   ) summer kanal 0-31
[  24] bs    s-1    ,hv    r-1     )
[  25] bs   (5)        IOB t-1     )
[  26] pp   ()      ,hv    21      ) p = indhopsadresse
[  27] sy    64     V  NZ          vr hvis FEJL
[  28] arn   -25    ,hv    -29     hent en konstant, hop til HJÆLP
[  29] sy    29     ,sy    60      )
[  30] sy    54     ,sy    53      ) skriv: FEJL
[  31] sy    33     ,sy    35      )
[  32] gr    -512      MOB         gem sum i 512, overløb i B
[  33] ly    33     ,vy    -496    vent, vælg strimmel input
[  34] pmn   64     XD IZA         )
[  35] tl    -7     ,ly    r1      )
[  36] pi              LZA t508    ) primitivt indlæseprogram
[  37] xr           X  IZB         )
[  38] hv    35        LZB         )
[  39] gr    41        MRC t-1     )
```

Ovenstaaende gælder for standard-GIER med hp-knap spærret med by0=1.

For hp-knap spærret med by3=1:

```
[   2] gk    1      ,vy    r79
[  33] ly    33     ,vy    80
```

[Testprogram no. 1.Topsøes test of the drum.
The program writes pseudo- random numbers on the tracks 1-319(the same number on all tracks) and compares the number that has been read with the written number.After each comparison for all the tracks is typed ok on the typewriter. By errors is typed track number,word number, the written word and the word that has been read.Bit 40 and 41 do not contain random numbers. KA=1 effects simultaneous computation by writing, KB=1 effects simultaneous computation by reading.Output on the typewriter can be removed by setting by=0 after the testprogram is read in.First instruction to be executed is zqLKB.
The drum parity check should be removed during run by the switch in GIER.]

```
i=41
[  41] zqLKB                    ;stop if KB=1
[  42] vy  16                   ;choice of typewriter
[  43] pa r8 t0                 ;choice of start track
[  44] pa r3 t-305              ;choice of start address for random  numbers
[  45] hs r31                   ;jump to sequ. for random  numbers
[  46] tk 1 ,it 1               ;shift
[  47] gr -305,hs r29           ;store random  numbers from 720 and jump
[  48] tk -30,ac (r-1)          ;shift and add
[  49] it(r-2),bs -265          ;is repeated 40-
[  50] hh r-5                   ;times
[  51] vk 0 t1                  ;choice of track
[  52] sk -304                  ;write track
[  53] hv r5 NKA                ;jump if KA=0
[  54] pa r2 t50                ;put in 50 as counting number
[  55] hs r21 t29               ;compute 50-
[  56] bt 50 t-1                ;random  numbers simultaneously-
[  57] hv r-2                   ;with writing
[  58] vk(r-7),lk -504          ;choose track and read until 520
[  59] hv r5 NKB                ;jump if KB=0
[  60] pa r2 t50                ;place 50
[  61] hs r15                   ;compute 50-
[  62] bt 50 t-1                ;random  numbers simultaneously-
[  63] hv r-2                   ;with reading
[  64] pa r3 t-305              ;choice of start address for random  numbers
[  65] pa r3 t-505              ;choice of start address for read numbers
[  66] vk (r-15)                ;wait for the drum
[  67] arn -305 t1              ;take the written
[  68] sr -505 t1               ;subtract the read
[  69] gr r15,arn r15           ;store the difference and fetch it again
[  70] hv r15 NZ                ;jump to typeout if not zero
[  71] it (r-3),bs -465         ;this is repeated-
[  72] hv r-5                   ;40 times
[  73] it (r-22),bs 319         ;if track number < 319
[  74] hv r-23                  ;jump back
[  75] hv r42                   ;else jump to typeout of ok
[  76] pm r7,mln r5             ;RAND-1,subroutine for random  numbers
[  77] tk 8,gm r6
[  78] ar r5
[  79] sr r5 LO
[  80] gr r3,hr s1              ;jump back
[  81] gtn p,dln r434            ;constant
[  82] udn (p511),gc -1         ;constant
[  83] sn p206 X t-228          ;constant
[  84] qq
[  85] sy 64,ud r39             ;type CR and count down in 120
[  86] arn(r-35) D            ;fetch track number
[  87] hs r17                 ;jump to decimal typeout
[  88] qq
[  89] arn(r-21) D            ;fetch word number relatively


[  90] sr -504 D              ;find word number
[  91] hs r13                 ;jump to decimal typeout
[  92] qq
[  93] sy 64                  ;type CR
[  94] arn(r-27),tk 1         ;fetch the written word and-
[  95] sy 1 V LT              ;write-
[  96] sy                     ;it-
[  97] hh r-3 NZ              ;until the rest is only o-es
[  98] sy 64                  ;write CR
[  99] arn(r-31),tk 1         ;fetch the read word and-
[ 100] sy 1V LT               ;write it-
[ 101] sy                     ;until the rest is only o-es
[ 102] hh r-3 NZ              ;
[ 103] hvr-32                 ;jump to repeat
[ 104] ck 10,tk 30            ;decimal typeout
[ 105] ck -10,pt r9
[ 106] dk r9 X
[ 107] pp 4,mln r9
[ 108] pp p-1,tk 20
[ 109] ar r5 V LZ
[ 110] pt r4 t16
[ 111] gt r , sy
[ 112] bs p510,ud r-2
[ 113] bs p , hh r-6
[ 114] hr s1,qq 16            ;jump back
[ 115] qq9  t-241
[ 116] qq  IZA
[ 117] sy 38 ,sy 34           ;type ok
[ 118] sy ,sy                 ;type two spaces
[ 119] sy ,sy                 ;type two spaces
[ 120] bt 12 t-1              ;up until 12 times
[ 121] hv 43                  ;jump to repeat
[ 122] pa r-2 t12             ;else place 12 as counting number
[ 123] sy 64 ,hv 43           ;write CR and jump to repeat
[ 124] pa r-4 t-1             ;count down to second to the last
e41
```

[Testprogram no. 2, typeout of bit-configuration or bit-pattern.Set the TYPEWRITER to the shortest distance between the lines.
Program is stored from location 41 to 89.The first instruction to be executed is zq LKB.The program will stop after 20 lines have been typed if KB=1.The typeout will be repeated by pressing START-button.]

```
i=41
[  41] zq LKB                   ;stop if KB=1
[  42] vy16 ,hs 50              ;choice of typewriter and jump
[  43] qq 70 t1                 ;constant and counter
[  44] bt 19 t-1                ;is performed-
[  45] hh r-3                   ;20 times
[  46] zq  LKB                  ;stop if KB=1
[  47] pa r-3 t19               ;set number of words
[  48] pa r-5 t70               ;set start address
[  49] hh r-7                   ;go to repeat
[  50] arn(s1) IPC              ;typeoutprogram.Fetch a word
[  51] sy 64 , sy 60            ;type CR and upper case
[  52] sy 6, tk 1               ;type [ and shift
[  53] sy 57 V LT               ;type I if sign is-
[  54] sy                       ;else type space
[  55] bt 39 t-1                ;is performed
[  56] hh r-4                   ;40 times
[  57] pa r-2 t39               ;set number of bits
[  58] sy 57 V LPA              ;type I if A-mark
[  59] sy                       ;else type space
[  60] sy 57 V LPB              ;type I if B-mark
[  61] sy                       ;else type space
[  62] sy 7,hr s1               ;type ] and jump back
[  63] qq                       ;empty
[  64] qq
[  65] qq
[  66] qq
[  67] qq
[  68] qq
[  69] qq
[  70] qq s1  IB t1             ;1 word in bitpattern
[  71] qq p3 IC t3
[  72] qq (p7) IKC t7
[  73] qqn (p15) IOC t15
[  74] zqn (p31) IRC t31
[  75] srn (p63) MRC t63
[  76] scn (p127) LRC t127
[  77] nkn (p255) D LRC t255
[  78] ppn (p511) VD LRC t511
[  79] udn (p-1) XVD LRC t-1
[  80] udn (p-1) XVD LRC t-1
[  81] udn (r-2) XVD LRA t-2
[  82] udn (-4) XVD LR t-4
[  83] udn -8 XVD LQ t-8
[  84] ud -16 XVD LT t-16
[  85] zl -32 XVD L t-32
[  86] hh -64 XVD N t-64
[  87] hv -128 XVD t-128
[  88] pc -256 XV t-256
[  89] pa -512 X t-512          ;last word in bitpattern
e41
```

[Testprogram no. 3.Typeout of track no. 0 in bit-pattern. Set the TYPEWRITER to double distance between the lines.
The testprogram is stored in locations 41 to 63,it uses the locations 64 to 103 for storing. The first instruction to be performed is zq LKB.The run can be repeated by pressing the START-button.]

```
i=41
[  41] zq LKB                   ;stop if KB=1
[  42] vk 0,lk 64               ;choose and read track no. 0
[  43] vy 16,vk 0               ;choose typewriter and wait for the drum
[  44] sy 62,hs 51              ;black ribbon and jump to typeoutprogram
[  45] qq 64 t1                 ;constant and counter
[  46] bt 39 t-1                ;is performed-
[  47] hh r-3                   ;40 times
[  48] pa r-3 t64               ;set start address
[  49] pa r-3 t39               ;set number of words
[  50] zq 0,hv r-7              ;stop and go to restart
[  51] arn (s1) IPC             ;typeoutprogram.Fetch a word
[  52] sy 64,sy 60              ;type CR and upper case
[  53] sy 6,tk 1                ;type [ and shift
[  54] sy 57 V LT               ;type I if sign is-
[  55] sy 4                     ;else type =
[  56] bt 39 t-1                ;is performed-
[  57] hh r-4                   ;40 times
[  58] pa r-2  t39              ;set number of bits
[  59] sy 57 V LPA              ;type I if A-mark
[  60] sy 4                     ;else type =
[  61] sy 57 V LPB              ;type I if B-mark
[  62] sy 4                     ;else type =
[  63] sy 7 ,hr s1              ;type ] and jump back
[  64] qq                       ;empty
e41
```

[Testprogram no. 4.Test of the fast memory or core-store.The content of M is stored in a location and is read back.If the read is equal to the written, the program will continue with the next location.The test is performed for the locations 60 - 1022.If KB=1 the program will stop and a new content can be placed in M.Position 40 and 41 in a location are not included in the test.By errors <f> is typed on the typewriter every time a content of a location is not hit right.If KA=1 GIER will stop while the number of the location that was hit wrong is placed in the in-register, and the difference between the read and the written is placed in the R-register.By restart or if KA=0 the program will go on testing the next location.The first instruction is zq LKB.]

```
i=41

[41] zq 0  LKB                  ;stop if KB=1
[42] gm 43 V                    ;store M in location no. 43
[43] qq 0                       ;empty
[44] gm 59 M t1                 ;store M in running location
[45] arn 43                     ;add content of location no. 43
[46] sr (44)                    ;subtract running location
[47] hv 52 NZ                   ;jump if R≠0
[48] arn 44                     ;add content of location no. 44
[49] ca 1022                    ;if number of running location is 1022
[50] pa 44 t59                  ;set 59 as address in location no. 44
[51] hv 41                      ;go to restart
[52] vy 16, sy 54               ;choose typewriter and type f
[53] pi (44)                    ;place no. of location in in-register
[54] zq 0 LKA                   ;stop if KA=1
[55] bt 40 t-1                  ;until 40 times
[56] hv 48                      ;jump
[57] pa 55 t40                  ;else place 40 as address in location 55
[58] sy 64 , hv 48              ;type CR and jump
[59] qq 0                       ;empty

e41
```

[Testprogram no. 5. Test of track no. 0.
The program reads drum track no. 0 to the locations 128-167 in the core store and compares for each location with the correct content of track no. 0 , that is read in to the locations 70-109 after the program.If there is no error in track no. 0 the message ok will be typed.If there is an error in one or more locations, the program will stop after having typed -i- for errors in positions 0-39 and -m- for errors in the mark-positions. The content of the wrong location can be observed in the R-register and the address of the wrong location in the s-register (read modulo 128). In the M-register is -by errors in pos. 0.39 - placed the difference from the subtraction and - by errors in the mark-bits - the correct mark-bits in position 8 and 9.]

```
i=41

[41] zq  LKB
[42] vy 16 , sy 58             ;the typewriter in lower case
[43] zq LKB
[44] vk  0, lk 128             ;read track 0 to loc. 128  and on
[45] vk ,pi                    ;wait for the drum
[46] arn 69 IRC +1             ;correct run cellcontent to R (markbits indic.)
[47] sr 127  +1                ;subtract next variable location
[48] gi 66 , pp                ;correct marks to location no. 66
[49] hv 60 NZ                  ;jump if R ≠ 0
[50] arn (47) IRC              ;read marks indic.
[51] arn 66 IK                 ;correct marks to R, read marks to p
[52] ncp , hv 63               ;jump if wrong marks
[53] arn 47,nc 167             ;typeout if 40 locations are tested
[54] hh 45                     ;or jump back
[55] sy 64                     ;type CR
[56] sy 38, sy 34              ;type ok
[57] pa 46  +69                ;replace address number
[58] pa 47  +127               ;replace address number
[59] hv 43
[60] sy 57  X                  ;type -i- ,difference to M
[61] arn (47) , ps(47)         ;wrong cellcontent to R,address to s
[62] zq  ,hv 50
[63] sy 36  X                  ;type -m- ,correct marks to M pos. 8 and 9
[64] arn(47), ps(47)           ;wrong cellcontent to R,address to s
[65] zq , hv 53
[66] qq                        ;location used for storing during run
```

[From location 70 to 109 is now placed the correct content of track no. 0. In this version it is assumed , that the HP-button is blocked by position 0 in the by-register.For a GIER, where the HP-button is blocked by pos. 3 in the by-register,the program can be used, if you before the run change the right half-word instructions in loc. 72 and 103 to vy r79 and vy 80]

```
i=70
[70]  qq
[71]  it 1,pa 10
[72]  gk 1, vy r-497              ;gk 1,vy r79 for HP blocked by pos. 3 in by-reg.
[73]  gi 2  IPC
[74]  gm 3  MPC
[75]  vk 32  IOB t-1
[76]  gr 4  MOB
[77]  tl -39 , pm 24
[78]  gr 7, tln -19
[79]  arn 2, cm -1
[80]  pmfn r  IRB
[81]  pi  66  IZA t-511
[82]  hh  17  NZA
[83]  gp  8 , pp  38
[84]  vk  319 t-1
[85]  sk -24  t-40
[86]  bt  23  t-1
[87]  hv  14,  pp  294
[88]  gs  6 , gin  9
[89]  vk  25 , lk -64
[90]  vk  p , lk -73
[91]  vk(5) ,lk -124
[92]  ps  41, ud  5
[93]  ps  s-1,  ar s-125
[94]  bs  s-1,hv r-1
[95]  bs (5)  IOB t-1
[96]  pp  () , hv 21
[97]  sy  64 V NZ
[98]  arn -25, hv -29
[99]  sy  29, sy  60
[100] sy  54, sy 53
[101] sy  33, sy  35
[102] gr -512  MOB
[103] ly 33, vy -496              ;ly 33,vy 80 for HP blocked by pos. 3 in by-reg.
[104] pmn  64 XD IZA
[105] tl  -7 ,ly r1
[106] pi   LZA t508
[107] xr  X  IZB
[108] hv  35  LZB
[109] gr  41  MRC t-1
e41
```

[Testprogram no. 6.Test of the core-store including marks.
The program examines the whole core-store exept for the locations that are occupied by the program itself.
At first zeroes are send to a location and the answer is checked, then ones are send to the same location and the answer is checked.

If stop in location 44 (r1=45): R0-39 ≠ 0, existing ones are wrong.

If stop in location 46 (r2=47): R40-41 ≠ 0, existing ones are wrong.

If stop in location 50 (r1=51): R0-39 ≠ 0, to position(s) containing a one and having a zero on the right side a zero has been send from the store instead of a one.

If stop in location 52 (r1=53): R=40.41 ≠ 1.1 , existing zeroes are wrong.

With LKB=1 the program will stop after having examined the core store from location 58 to 1023 and further on from location 0 to 40.In the indicator, the address of the location that is tested, can be observed.]

```
i=41

[41] zq LKB
[42] grn 57  M +1              ;zero to variable location
[43] arn(42) ,pi (42)
[44] zq NZ                     ;stop if R0-39 ≠ 0
[45] qq  V NC
[46] zq                        ;stop if R40.41 ≠ 0.0
[47] srn 57                    ;constant is subtracted
[48] gr (42) MC                ;ones to variable location
[49] arn 57 , ar (42)          ;constant to R,content of location is added
[50] zq NZ                     ;stop if R0-39 ≠ 0
[51] qq  V LC
[52] zq                        ;stop if R40.41 ≠ 1.1
[53] arn 42 ,nc 40
[54] hv 42                     ;examine the next (variable) location
[55] pa 42 +57                 ;replace the address 57 in location 42
[56] hv 41                     ;restart with location no. 58
[57] qq IB                     ;constant( 1 bit in position 39)

e41
```

[Testprogram no. 7. Topsøes test of the floating point operations: AR F,SR F, MK F and DK F.
The testprogram operates in 6000 cycles.In each cycle two random,floating point numbers,a and b, are generated by means of the subroutines RAND-1 and RAND-2.The four floating point operations are then carried out on the numbers a and b, and the four results are added(as fixed points numbers) to four sum cells.
After 6000 cycles a line is printed on the typewriter,containing the number 6000 and the four sum cells in octal notation.The four sum cells are then cleared and 6000 new cycles are calculated,etc.If KB=1 the program will stop after 6000 cycles, and can be repeated by pressing the NORMAL START-button.

The correct output is (only the numbers):

```
6000   35271407753020   36242706607000   66043725657530   15577777733744
6000   35271407753020   36242706607000   66043725657530   15577777733744
etc.
test of:    add.            sub.             mult.            div.
```

Calculation of 6000 cycles takes about 35 sec.The value of 6000 is stored in cell 114 as an integer in pos. 39.It may be changed if required.
If KA is set to 1, output is obtained after each cycle.

The program is stored from cell 41 to 136.he first instruction to be executed is zq LKB.]

```
i=41

[ 41]  zq LKB                      ;stop if KB=1
[ 42]  vy 16,hv r4                 ;typewriter output,go to 46
[ 43]  qq 0
[ 44]  qq 0
[ 45]  zq LKB                      ;stop if KB=1
[ 46]  grn r71 [117]               ;sum:=0
[ 47]  grn r71 [118]               ;sum:=0
[ 48]  grn r71 [119]               ;sum:=0
[ 49]  grn r71 [120]               ;sum:=0
[ 50]  hv r19                      ;go to 69
[ 51]  qq 0
[ 52]  qq 0
[ 53]  qq IZA                      ;parameter
[ 54]  gcn s9  t320                ;parameter
[ 55]  pi      t-65                ;from 55 to 66 is placed -
[ 56]  arn r-2,gr r-13             ;a subroutine for printing -
[ 57]  dln r-14,tk 20              ;out the counter in location -
[ 58]  pi  64  NZ t-65             ;115 in decimal notation.
[ 59]  hv  r3   NZ
[ 60]  hv  r2   NTB
[ 61]  sy  16,hv  r2
[ 62]  gt  r ,sy
[ 63]  gm  r-11,pm  r-20
[ 64]  dln  r-11,gr  r-21
[ 65]  hr s2   LZ
[ 66]  pm  r-14,hv  r-9
[ 67]  qq         t256             ;parameter
[ 68]  qq         IB               ;parameter
[ 69]  grn  r-25                   ;zeroes to cell 44
[ 70]  arn  r39,gr  r65            ;mge start-random from 109 to 135
[ 71]  grn  r44                    ;counter:=0
[ 72]  pp  2,pp  p-1               ;p:=2 , p:=p-1
[ 73]  grn  p121,hs  r51           ;cell:=0 , go to RAND-2
[ 74]  qq  20                      ;parameter
[ 75]  sr   10 D                   ;-10
[ 76]  ga   p121,hs  r52           ;set exp. ,go to RAND-1
[ 77]  tk   2    ITA               ;set sign
[ 78]  ck   -6,tl  -6              ;Roo:=0
[ 79]  ac   p121,arn  r-12         ;store , fetch 1
[ 80]  tk   1    LTA               ;if neg. then -2
[ 81]  ac   p121                   ;add to cell
[ 82]  bs   p,hh  r-10             ;repeat
[ 83]  pp   4,pp  p-1              ;p:=4 , p:=p-1
[ 84]  arfn  r37,udf  p110         ;fetch b , execute
[ 85]  grf  r38                    ;store in 123
[ 86]  arn  r37,ac  p117           ;add to cell
[ 87]  bs   p,hh  r-4              ;repeat
[ 88]  hsn  r5 X LKA t52           ;check output
[ 89]  arn  r-21,ar  r26           ;count
[ 90]  gr   r25,sr  r24            ;- limit
[ 91]  hv   r-19   LT              ;repeat
[ 92]  psn  r43                    ;final output
[ 93]  sy   64,arn  r22            ;CAR RET,counter to R-reg.
[ 94]  hs   r-39 X  t59            ;print counter
[ 95]  qq   257  t-144             ;parameters
[ 96]  sy   ,sy                    ;2 SPACES
[ 97]  pp   4,pp  p-1              ;p:=4 , p:=p-1
[ 98]  arn  p117,tl  -10           ;fetch number , shift
[ 99]  ga   r17,sr  r17            ;delete bits 0-9
[100]  tl   3,ca                   ;shift , if zero
[101]  sy   16,hv  r2              ;then print zero
[102]  ga   r14,sy (r14)           ;store and print
[103]  bt   13  t-1                ;repeat 13 times
[104]  hv   r-5
[105]  pan  r-2  X  t13            ;reset address in bt-instruction
[106]  sy   ,sy                    ;2 SPACES
[107]  bs   p,hh  r-10             ;repeat
[108]  hr   s1                     ;exit ( to 136)
[109]  bs   p104  V  t-333         ;start random
[110]  dkf  r11                    ;/b
[111]  mkf  r11                    ;Xa
[112]  srf  r10                    ;-a
[113]  arf  r9                     ;+a
[114]  qq   (s)   XV  LT           ;6000 = limit
[115]  qq                          ;counter
[116]  qq                          ;storage of address
[117]  qq                          ;storage of div.
[118]  qq                          ;storage of mult.
[119]  qq                          ;storage of subtr.
[120]  qq                          ;storage of add.
[121]  qq                          ;storage of b
[122]  qq                          ;storage of a
[123]  qq                          ;storage of result
[124]  hs   r4  t44                ;RAND-2
[125]  xr   ,mkn  s1
[126]  mb   511 D
[127]  hr   s2
[128]  pm   r7,mln  r5             ;RAND-1
[129]  tk   8,gm  r6
[130]  ar   r5
[131]  sr   r3  LO
[132]  gr   r3,hr  s1
[133]  gtn  p,dln  r434
[134]  udn (p511),pc  -1
[135]  qq                          ;age of start random
[136]  hv   45                     ;go to 6000 new cycles
e41
```

[Testprogram no. 8. Test of drum track transfer time.
The time for execution of an LK instruction is tested by the program.If KA is set to 1, the transfer time for all tracks is printed out.If KA=0 there will only be output for tracks with a transfer time different from 20 ms. The output is in binary notation for the transfer times as well as for the track numbers.

After the end of a test run (output <slut> on the typewriter), the test can be repeated by pressing the NORMAL START-button. If the qq instruction in location 57 is changed to a zq instruction,the actual transfer time can after stop in location 57 be observed in the R-register in microseconds with unit in pos. 19.
First instruction to be executed is zq LKB in location 41.]

```
i=41
[41]  zq LKB                    ;stop if KB=1
[42]  vy 16,sy 64               ;select typewriter,CR
[43]  sy 58,sy 36               ;lower case,m
[44]  sy 18                     ;s
[45]  bt 10 t-1                 ;write 10 SPACE
[46]  sy  ,hv 45
[47]  sy 19,sy41                ;t,r
[48]  sy 49,sy 51               ;a,c
[49]  sy34,sy 64                ;k,CR
[50]  bt 320 t-1                ;select -
[51]  vk  -1  V  t1             ;- tracks 0-319
[52]  hv 75                     ;go to 75 if track no. =319
[53]  lkn -512                  ;read track, zeroes to R
[54]  hk 57                     ;if transfer finished then go to 57 -
[55]  ar 74                     ;else add 113 microseconds
[56]  hv 54                     ;and go to 54
[57]  qq                        ;empty
[58]  nc 20 NKA                 ;output for all tracks if KA=1 -
[59]  ps 66 V                   ;- else only output for tracks -
[60]  hv 50                     ; - with transfer time ≠ 20 ms
[61]  tk 1                      ;shift 1 position to the left
[62]  sy 1  V LT                ;print 1 if sign
[63]  sy 38                     ;else print o
[64]  bt 9 t-1                  ;and repeat 9 times -
[65]  hv 61                     ;- from 61
[66]  hv s1
[67]  sy  ,sy                   ;2 SPACE
[68]  pa 64 t9                  ;reset address in loc. 64
[69]  arn (51) D                 ;track no. to R
[70]  hs 61                     ;print track no.
[71]  pa 64 t9                  ;reset address in loc. 64
[72]  sy  64                    ;CR
[73]  hv  50                    ;go to new track
[74]  qq  t113                  ;number of microseconds
[75]  sy  18                    ;s
[76]  sy  35,sy  20             ;l,u
[77]  sy  19,sy  64             ;t,CR
[78]  zq                        ;stop
[79]  pa  45  t10               ;reset loc. 45
[80]  pa  50  t320              ;reset loc. 50
[81]  pa  51t-1                 ;reset loc. 51
[82]  hv  41                    ;go to start
e41
```

[Testprogram no. 9. Test of the IBM-typewriter for input/output.

When the testprogram is read into GIER, a correct run will be, that every character, that is manually typed on the typewriter, is typed out again 9 times by the program. If upper case is typed, a serie of case shifts will be done.

If KA=0, the characters are tested in lower case, if KA=1, they are tested in upper case.

First instruction to be executed is zq LKB.]

```
i=41
[41]  zq  LKB
[42]  vy 17, ly r3               ;typewriter input/output, read to 45
[43]  sy 60  LKA                 ;upper case if KA=1
[44]  sy 58  NKA                 ;lower case if KA=0
[45]  qq  0, ca 60               ;if upper case is input-character
[46]  hv r5                      ;then go to caseshift
[47]  pa r1  t9                  ;reset address in loc. 48
[48]  bt 9  t-1                  ;9 times -
[49]  sy  (r-4),hv r-1           ;- print the read character
[50]  sy 64,hh 42                ;type CR, go to new character
[51]  sy 60, sy 58               ;case shift
[52]  sy 60, sy 58
[53]  sy 60, sy 58
[54]  sy 60, sy 58
[55]  sy 60,  LKA                ;end in upper case if KA=1
[56]  hh 42                      ;go to new character
e41
```

[Testprogram t10 (ferrittest)

The program test reading and writing in cells 512-1023 in the fast memory. If a cell in this interval is mistaken for another cell in the same interval there will be no errorreaction , because the program use the same bitpattern in all cells during one testcycle.The program is stored from cell 41 to 131.

INPUT OF PROGRAM

The program is published in a condensed version (a B-tape) to be read into the machine by the basic inputprogram on channel 0,cells 34-39.After the input a sumcheck of the program is performed.If this check is ok. the program writes <CR,t 10 > on the typewriter with black ribbon, otherwise the programname will be typed in red,and the program stops in cell 49.

TESTDIGITS

After reading in and sumcheck the program will stop in cell 51 if KB=1.During this stop the registers R and In contains zeroes.If no bits are inserted the program itself will generate the testdigits for the run.If bits are inserted in R or In these bits will serve as testdigits during the run.In In only RA and RB gives sense.

BUILD IN TESTDIGITS

When the build in testdigits are used , 42 consecutive bits is taken from a group of bits containing:

$$\underbrace{1,1,1,1,\ldots.,1}_{42\text{ bits}},\underbrace{0,0,0,0,\ldots.,0}_{42\text{ bits}},\underbrace{1,1,1,1,\ldots.,1}_{41\text{ bits}}$$

This means that each cell is automatically tested with 84 different (but known) bitpatterns.

OPERATORS TESTDIGITS

If the operator choose the testdigits each cell will be tested 81 times with the choosen bitcombination.The operator must insert the bits in R0-39 and the marks in In[RA,RB].

ERROR REACTIONS

During the running of the program , the normal condition of KA and KB is KA=KB=0.In this case. there will be no error-output before one complete run is terminated.A complete run means writing and reading in cells 512-1023 of either the 84 generated bitpatterns or 81 times with the operators bits.

NO ERROR: If no error is registrated the run will terminate with the typed message <CR,ok> and a hoot at 1000 c/s in 5 seconds.

ERROR: If an error is registrated the run will terminate with the typed message <CR,-> and the frequency of the hoot will alternate between 950 c/s and 1100 c/s.

KA=1 will cause the program to stop when an error is detected in the bits 0-39 or the marks in the cell in question.There is two possible stoppoints in the program:

1) Stop with r1=122 [0001111010] indicates error in bits 0-39. The testbits used will be found in M.Bits in R set to 1 are errors.Depressing the START-button two times while the STOP-button is depressed will transfer the adress of the cell to p and the cellcontent to R0-39.Depressing START will continue the test.
2) Stop with r1=125 [0001111101] indicates error in the marks. The testmarks is stored in R8-9 and the marks of the cell in In[RA,RB].The adress of the cell is transferred to p. Depressing START will continue the test.

Setting KB=1 when the machine is stopped in an error , and depressing START , will cause cycling in a loop as long as KB=1.In this loop th content of M0-39 and In[RA,RB] is stored in the cell , and the content of the cell is read to R0-39 and In[QA,QB].]

```
[Test of the fast memory,cell 512-1023]
i=   41
[  41] vy   16          ,psn  132
[  42] ps   s-1         ,ar   s
[  43] bs   s-41        ,hv   r-1 [42]   ;form checksum of the program
[  44] sy   62          ,tk   1
[  45] sy   29            NZ             ;if errorsum then redshift
[  46] sy   64          ,sy   19
[  47] sy               ,sy   1
[  48] sy   16          ,sy   62         ;outtext(outcr,≮<t 10≯,blackshift)
[  49] zq                 NZ             ;if errorsum then stop
[  50] pin                               ;R:=In:=0
[  51] zq                 LKB            ;if KB then stop for manual testdigits
[  52] gr   r57 [109]     M              ;controlcell 40,41 := 0,0
[  53] pa   r35 [88]            t1       ;manualdigits:=true
[  54] gi   r2 [56]     V  LZ            ;if R=0 then c56adr:=In / goto c56
[  55] pa   r18 [73]    ,hv   r36 [91]   ;error:=false / goto operators testdigits
[  56] nc               ,hv   r-1 [55]   ;if In≠0 then goto c55
[  57] pa   r31 [88]    ,pa   r16 [73]   ;manualdigits:=false / error:=false
[  58] pm    D                           ;M:=0
[  59] pi               ,hs   101        ;In:=0 / test
[  60] pi   1           ,hs   101        ;In:=1 / test
[  61] pi   3           ,hs   101        ;In:=3 / test
[  62] pm   r37 [99]    ,hs   101        ;M 0-39 := all 1 / test
[  63] xrn              ,cl   -1         ;R:=M×2↑(-1)
[  64] hh   r-2 [62]    X  NZ            ;if M≠0 then begin M:=R;goto c62 end
[  65] pm   r34 [99]    ,pi   2          ;M 0-39 := all 1 / In:=2
[  66] hs   101                          ;test
[  67] pi               ,hs   101        ;In:=0 / test
[  68] xr               ,tk   1          ;R:=M×2
[  69] pm   r30 [99]    ,cm   r31 [100]  ;if R≠-1 then begin M:=R;goto c71 end
[  70] hh   r-3 [67]    X                ;else goto c67
[  71] gr   r38 [109]   XV MRC           ;controlcell 40,41 := 1,1 / M:=R
[  72] hs   101                          ;test
[  73] ps               ,sy   64         ;s:=error / outcr
[  74] can  s           ,hv   r2 [76]    ;if -,error then goto c76
[  75] sy   32          ,hv   r2 [77]    ;else writetext(≮<-≯) / goto c77
[  76] sy   38          ,sy   34         ;writetext(≮<ok≯)
[  77] pa   r9 [86]             t9
[  78] pa   r5 [83]             t511     ;initialize hootduration
[  79] pa   r2 [81]             t7       ;initialize frequency-cycle
[  80] ar   s128        D                ;Radr := (if error then 8 else 0)+128+Radr
[  81] bt   7                   t-1      ;for i:=7 step -1 until 1 do
[  82] hv   r-2 [80]                     ;goto c80
[  83] bt   511                 t-1      ;for j:=511 step -1 until 1 do
[  84] hv   r-5 [79]                     ;goto c79
[  85] ns   s           ,ps   s          ;s:=-s
[  86] bt   9                   t-1      ;for k:=9 step -1 until 1 do
[  87] hv   r-9 [78]                     ;goto  c78
[  88] bs                                ;if manualdigits then
[  89] hv   r-38 [51]   X                ;goto newtest(Operatorsdigits)
[  90] hv   r-40 [50]                    ;else goto newtest(buildindigits)
[  91] pm   r8 [99]     ,cm   r9 [100]   ;Operatorsdigits:
[  92] hv   r2 [94]                      ;if R≠-1 then goto c94
[  93] gr   r16 [109]     MRC            ;controllcell 0-39 := R ;40,41:=RA,RB
[  94] pa   r2 [96]     X       t80      ;M:=R / for i:=1 step 1 until 81 do
[  95] hs   101                          ;test
[  96] bt                          t-1
[  97] hv   r-2 [95]
[  98] hv   r-25 [73]                         ;test finished
[  99] udn (p-1)          XVDLRC t-1          ;constant 0-39 := all 1
[ 100] qq   -512                              ;procedure test;begin
[ 101] pa   r3 [104]               t511       ;entry to test
[ 102] pa   r1 [103]                          ;for j:=1023 step -1 until 512 do
[ 103] gm                    MRC t-1          ;cell j 0-39:= M cellj 40-41 :=RA,RB
[ 104] bt   511                    t-1
[ 105] hv   r-2 [103]
[ 106] pa   r12 [118]              t511       ;for i:=1023 step -1 until 512 do begin
[ 107] pa   r7 [114]
[ 108] grn  r1 [109]      XV                  ;controlcell 0-39 := 0 / R:=testdigits
[ 109] qq
[ 110] sc   r-1 [109]     X                   ;controlcell:=-testdigits / M:=testdigits
[ 111] gi   r9 [120]                          ;c120adr:=In
[ 112] pi                 ,arn  r-3 [109]     ;In:=0 / R:=-testdigits
[ 113] srn  r-4 [109]        LC               ;if R=-1 then R:=-(-testdigits)
[ 114] ar                    IRC t-1          ;R:=R+cell[i] / In[RA,RB]:=cell i 40,41
[ 115] hv   r6 [121]         NZ               ;if R≠0 then goto errorincell
[ 116] gi   r1 [117]      ,arn  r4 [120]      ;c117adr := In / R:= old In
[ 117] nc                 ,hh   r6 [123]      ;if newmarks≠oldmarks then goto errorinmarks
[ 118] bt                          t-1        ;one cell tested
[ 119] hv   r-7 [112]
[ 120] pi                 ,hr   s1            ;In:=Original marks / end of procedure test;
[ 121] zq                    LKA              ;errorincell: if errorincell∧KA then stop
[ 122] pp  (r-8)[114]     ,arn  p             ;p:=adress of wrong cell / R:=wrong cell
[ 123] hv   r2 [125]      ,pp  (r-9)[114]     ;goto seterror / errorinmarks: p:=adress of wrong
[ 124] zq                    LKA              ;if error in marks ∧KA then stop
[ 125] pa   r-52 [73]              t8         ;seterror: error:=true
[ 126] pi  (r-6)[120]     ,hv   r3 [129]      ;In:=original marks
[ 127] gm   p                MRC              ;testdigits to cell
[ 128] arn  p                IQC              ;cell to R
[ 129] hv   r-2 [127]        LKB              ;if KB then cycle in 127-128
[ 130] pi  (r-10)[120]    ,hv   r-12 [118]    ;In:=Original marks / continue the test
[ 131] gmn  -319          X  LP  t488         ;checksum of program
s
```

[Testprogram t11 (ferrittest)

The program test reading and writing in cells 0-511 in the fast memory. If a cell in this interval is mistaken for another cell in the same interval there will be no errorreaction , because the program use the same bitpattern in all cells during one testcycle.The program is stored from cell 513 to 603 .

INPUT OF PROGRAM

The program is published in a condensed version (a B-tape) to be read into the machine by the basic inputprogram on channel 0,cells 34-39.After the input a sumcheck of the program is performed.If this check is ok. the program writes <CR,t 10 > on the typewriter with black ribbon, otherwise the programname will be typed in red,and the program stops in cell 521.

TESTDIGITS

After reading in and sumcheck the program will stop in cell 523 if KB=1.During this stop the registers R and In contains zeroes.If no bits are inserted the program itself will generate the testdigits for the run.If bits are inserted in R or In these bits will serve as testdigits during the run.In In only RA and RB gives sense.

BUILD IN TESTDIGITS

When the build in testdigits are used , 42 consecutive bits is taken from a group of bits containing:

$$\underbrace{1,1,1,1,....,1}_{42\text{ bits}},\underbrace{0,0,0,0,....,0}_{42\text{ bits}},\underbrace{1,1,1,1,....,1}_{41\text{ bits}}$$

This means that each cell is automatically tested with 84 different (but known) bitpatterns.

OPERATORS TESTDIGITS

If the operator choose the testdigits each cell will be tested 81 times with the choosen bitcombination.The operator must insert the bits in R0-39 and the marks in In[RA,RB].

ERROR REACTIONS

During the running of the program , the normal condition of KA and KB is KA=KB=0.In this case there will be no error-output before one complete run is terminated.A complete run means writing and reading in cells 0-511 of either the 84 generated bitpatterns or 81 times with the operators bits.

NO ERROR: If no error is registrated the run will terminate with the typed message <CR,ok> and a hoot at 1000 c/s in 5 seconds.

ERROR: If an error is registrated the run will terminate with the typed message <CR,-> and the frequency of the hoot will alternate between 950 c/s and 1100 c/s.

KA=1 will cause the program to stop when an error is detected in the bits 0-39 or the marks in the cell in question.There is two possible stoppoints in the program:

1) Stop with r1=594[1001010010] indicates error in bits 0-39. The testbits used will be found in M.Bits in R set to 1 are errors.Depressing the START-button two times while the STOP-button is depressed will transfer the adress of the cell to p and the cellcontent to R0-39.Depressing START will continue the test.
2) Stop with r1=597[1001010101] indicates error in the marks. The testmarks is stored in R8-9 and the marks of the cell in In[RA,RB].The adress of the cell is transferred to p. Depressing START will continue the test.

Setting KB=1 when the machine is stopped in an error , and depressing START , will cause cycling in a loop as long as KB=1.In this loop th content of M0-39 and In[RA,RB] is stored in the cell , and the content of the cell is read to R0-39 and In[QA,QB].]

```
[Test of the fast memory,cells 0-511]
i= -511
[-511] vy   16              ,psn  -420
[-510] ps   s-1             ,ar   s
[-509] bs   s511            ,hv   r-1 [-510]     ;form checksum of the program
[-508] sy   62              ,tk   1
[-507] sy   29                NZ                 ;if errorsum then redshift
[-506] sy   64              ,sy   19
[-505] sy                   ,sy   1
[-504] sy   1               ,sy   62             ;typetext(CR,<<t 11>>,blackshift)
[-503] zq                     NZ                 ;if errorsum then STOP
[-502] pin                                       ;R:=In:=0
[-501] zq                     LKB                ;if KB=1 then STOP for manual testdigits
[-500] gr   r57 [-443]        M                  ;controlcell 40,41:= 0,0
[-499] pa   r35 [-464]             t1            ;manualdigits:=true
[-498] gi   r2 [-496]       V  LZ                ;if R=0 then c528adr:=In / goto c528
[-497] pa   r18 [-479]      ,hv   r36 [-461]     ;error:=false / goto operators testdigits
[-496] nc                   ,hv   r-1 [-497]     ;if In≠0 then goto c527
[-495] pa   r31 [-464]      ,pa   r16 [-479]     ;manualdigits:= false / error:= false
[-494] pm    D                                   ;M:= 0
[-493] pi                   ,hs   -451           ;In:= 0 / test
[-492] pi   1               ,hs   -451           ;In:= 1 / test
[-491] pi   3               ,hs   -451           ;In:= 3 / test
[-490] pm   r37 [-453]      ,hs   -451           ;M0-39:= all 1 / test
[-489] xrn                  ,cl   -1             ;R:= M×2↑(-1)
[-488] hh   r-2 [-490]      X  NZ                ;if M≠0 then begin M:=R;goto c534 end
[-487] pm   r34 [-453]      ,pi   2              ;M0-39:= all 1 / In:=2
[-486] hs   -451                                 ;test
[-485] pi                   ,hs   -451           ;In:= 0 / test
[-484] xr                   ,tk   1              ;R:= M×2
[-483] pm   r30 [-453]      ,cm   r31 [-452]     ;if R≠-1 then begin M:=R;goto c543 end
[-482] hh   r-3 [-485]      X                    ;else goto c539
[-481] gr   r38 [-443]      XV MRC               ;controlcell 40,41:= 1,1 / M:=R
[-480] hs   -451                                 ;test
[-479] ps                   ,sy   64             ;s:= error / typetext(CR)
[-478] can  s               ,hv   r2 [-476]      ;if ¬,error then goto c548
[-477] sy   32              ,hv   r2 [-475]      ;else typetext(<<->>) / goto c549
[-476] sy   38              ,sy   34             ;typetext(<<ok>>)
[-475] pa   r9 [-466]              t9
[-474] pa   r5 [-469]              t511          ;initialize hootduration
[-473] pa   r2 [-471]              t7            ;initialize frequencycycle
[-472] ar   s128            D                    ;Radr:= Radr+128+(if error then 8 else 1)
[-471] bt   7                      t-1           ;for i:= 7 step -1 until 1 do
[-470] hv   r-2 [-472]                           ;goto c552
[-469] bt   511                    t-1           ;for j:= 511 step -1 until 1 do
[-468] hv   r-5 [-473]                           ;goto c551
[-467] ns   s               ,ps   s              ;s:= -s
[-466] bt   9                      t-1           ;for k:= 9 step -1 until 1 do
[-465] hv   r-9 [-474]                           ;goto c550
[-464] bs                                        ;if manualdigits then
[-463] hv   r-38 [-501] X                        ;newtest(operators testdigits)
[-462] hv   r-40 [-502]                          ;else newtest(buildin testdigits)
[-461] pm   r8 [-453]       ,cm   r9 [-452]      ;operators testdigits:
[-460] hv   r2 [-458]                            ;if R≠ -1 then goto c566
[-459] gr   r16 [-443]        MRC                ;controlcell0-39:= R , 40-41:=RA-RB
[-458] pa   r2 [-456]       X      t80           ;M:=R / for i:= 1 step 1 until 81 do
[-457] hs   -451                                 ;test
[-456] bt                          t-1
[-455] hv   r-2 [-457]
[-454] hv   r-25 [-479]                          ;test finished
[-453] udn  (p-1)           XVDLRC t-1           ;constant0-39= all 1
```

```
[-452] qq   -512                           ;procedure test;begin
[-451] pa   r3 [-448]          t511        ;entry to test
[-450] pa   r1 [-449]          t-512       ;for j:= 511 step -1 until 0 do
[-449] gm                   MRC t-1        ;cellj0-39:= M, 40-41:=RA,RB
[-448] bt   511                t-1
[-447] hv   r-2 [-449]
[-446] pa   r12 [-434]         t511        ;for i:= 511 step -1 until 0 do begin
[-445] pa   r7 [-438]          t-512
[-444] grn  r1 [-443]   XV                 ;controlcell0-39:= 0 / R:=testdigits
[-443] qq
[-442] sc   r-1 [-443]  X                  ;controlcell:= -testdogits / M:= testdigits
[-441] gi   r9 [-432]                      ;c592adr:= In
[-440] pi               ,arn  r-3 [-443]   ;In:= 0 / R:=-testdigits
[-439] srn  r-4 [-443]     LC              ;if R= -1 then R:=-(-testdigits)
[-438] ar                  IRC t-1         ;R:=R+cell i / In[RA,RB]:= cell i 40,41
[-437] hv   r6 [-431]      NZ              ;if R≠ 0 then goto error in cell
[-436] gi   r1 [-435]   ,arn  r4 [-432]    ;c589adr:= In / Radr:= old In
[-435] nc               ,hh   r6 [-429]    ;if newmarks≠ old marks then goto error in marks
[-434] bt                      t-1         ;one cell tested
[-433] hv   r-7 [-440]
[-432] pi               ,hr   s1           ;In:= original marks / end of procedure test;
[-431] zq                  LKA             ;error in cell: if error∧KA=1 then STOP
[-430] pp  (r-8)[-438]  ,arn  p            ;p:=adress of error / R:=errorcell
[-429] hv   r2 [-427]   ,pp  (r-9)[-438]   ;goto set error / error in marks: p:=erroradress
[-428] zq                  LKA             ;if error∧KA=1 then STOP
[-427] pa   r-52 [-479]        t8          ;set error: error:=true
[-426] pi  (r-6)[-432]  ,hv   r3 [-423]    ;In:= original marks
[-425] gm   p              MRC             ;send testdigits to cell
[-424] arn  p              IQC             ;read cell to R
[-423] hv   r-2 [-425]     LKB             ;if KB=1 then cycle in cells 599-600
[-422] pi  (r-10)[-432] ,hv   r-12 [-434]  ;In:= original marks / continue with next cell
[-421] gmn  -228        X  LP  t-296       ;checksum of the program
s
```

INDICATOR TEST

Before start KA, KB are set to 0, 1. The test is supplied with a number of loops, so if any error occurs, KA is set to one, and the program will run in the loop, where the error appeared. All stops indicate errors. Cellnumber is found by subtraction of one from the content of r_1.

```
41   zq            LKB
42   grn   3oo
43   pi    -1
44   gi    3oo
45   arn   3oo
46   sr    -1  D
47   zq            NZ
48   hv    43      LKA   loop A, test of flip-flops for "ones"
49   pi
5o   gi    3oo     MOC
51   arn   3oo
52   zq            NZ
53   hv    49      LKA   loop B, test of flip-flops for "zeroes"
54   pi    -1            t-2 inhibit changing all except RB
55   zq            NRB
56   pi    -2            t-3 inhibit changing all except RA
57   zq            NRA
58   hv    6o      LRC ⎫
59   zq                ⎪
6o   zq            LOA ⎪
61   zq            LOB ⎪
62   zq            LTA ⎪
63   zq            LTB ⎬ proving "L" indicator orders
64   zq            LPA ⎪
65   zq            LPB ⎪
66   zq            LQA ⎪
67   zq            LQB ⎭
68   gi    3oo ,arn  3oo
69   sr    3   D
7o   zq            NZ
```

96: gi 300, arn 300
rot

169: pi -512 t+511

```
 71   qq            MOC   R40-41 are set to zero
 72   ppn           IRB   dummy order , RB: = 0
 73   ppn           IRA   dummy order , RA: = 0
 74   zq            LRB
 75   zq            LRA
 76   qq            M     R40-41 are set to one
 77   ck    10      IRB   dummy order, RB: = 1
 78   ck    10      IRA   dummy order, RA: = 1
 79   zq            NRA
 80   zq            NRB
 81   hv    54      LKA   loop C, test of RA, RB
 82   pi    -4            t-5 inhibit changing all except QB
 83   zq            NQB
 84   pi    -8            t-9 inhibit changing all except QA
 85   zq            NQA
 86   hv    88      LQC
 87   zq
 88   zq            LOA
 89   zq            LOB
 90   zq            LTA  } proving "L and N" indicator orders
 91   zq            LTB
 92   zq            LPA
 93   zq            LPB
 94   zq            NRA
 95   zq            NRB
 96   gi          , arn    300
 97   sr    15  D
 98   zq            NZ
 99   gmn   400     IQB   dummy order, QB: = 0
100   gmn   400     IQA   dummy order, QA: = 0
101   zq            LQB
102   zq            LQA
103   qq            M
104   tk    2       IQB   dummy order, QB: = 1
105   tk    2       IQA   dummy order, QA: = 1
106   zq            NQB
107   zq            NQA
108   hv    82      LKA   loop D, test of QA, QB
```

```
109   pi    -16          t-17 inhibit changing all except PB
110   zq          NPB
111   pi    -32          t-33 inhibit changing all except PA
112   zq          NPA
113   hv    115   LPC  ⎫
114   zq               ⎪
115   zq          LOA  ⎪
116   zq          LOB  ⎪
117   zq          LTA  ⎬ proving "L" and "N" indicator orders
118   zq          LTB  ⎪
119   zq          NQA  ⎪
120   zq          NQB  ⎪
121   zq          NRA  ⎪
122   zq          NRB  ⎭
123   gi    300
124   arn   300
125   sr    63  D
126   zq          NZ
127   gmn   400   IPB    dummy order, PB: = 0
128   gmn   400   IPA    dummy order, PA: = 0
129   zq          LPB
130   zq          LPA
131   qq          M
132   tl    4     IPB    dummy order, PB: = 1
133   tl    4     IPA    dummy order, PA: = 1
134   zq          NPB
135   zq          NPA
136   hv    109   LKA    loop E, test of PA and PB
137   pi    -64          t-65 inhibit changing all except TB
138   zq          NTB
139   pi    -128         t-129 inhibit changing all except TA
140   zq          NTA
141   hv    143   LTC
142   zq
```

```
143  zq            LOA
144  zq            LOB
145  zq            NTA
146  zq            NTB
147  zq            NPA    proving "L" and "N" indicator orders
148  zq            NPB
149  zq            NQA
150  zq            NQB
151  zq            NRA
152  zq            NRB
153  gi   300
154  arn  300
155  sr   255   D
156  zq            NZ
157  qqn           ITB    TB: = 0
158  qqn           ITA    TA: = 0
159  zq            LTB
160  zq            LTA
161  arn  -1    D  ITB    TB: = 1
162  arn  -1    D  ITA    TA: = 1
163  zq            NTB
164  zq            NTA
165  hv   137      LKA    loop F, test of TA and TB
166  pi   -256            t-257 inhibit changing all except OB
167  zq            NOB
168  zq            NZB
169  pi   -512            t-511 inhibit changing all except OA
170  zq            NOA
171  zq            NZA
172  hv   174      LOC
173  zq
174  hv   176      LZC
175  zq
176  zq            NTA
177  zq            NTB    proving "L" and "N" indicator orders
178  zq            NPA
179  zq            NPB
180  zq            NQA
181  zq            NQB
182  zq            NRA
183  zq            NRB
```

```
184  gi   300
185  arn  300
186  sr   -1    D
187  zq             NZ
188  arn            IOB   dummy order, OB: = 0
189  qqn            IOA   dummy order, OA: = 0
190  zq             LOB
191  zq             LOA
192  arn  -1    D
193  ar   -512  D   IOB                  OB: = 1
194  ar   -512  D   IOA                  OA: = 1
195  zq             NOB
196  zq             NOA
197  hv   166       LKA   loop G, test of OA, OB
198  pi   -1
199  pi                   t-1
200  hv   202       LZC )
201  zq                 |
202  hv   204       LOC |
203  zq                 |
204  hv   206       LTC |
205  zq                  > proving "L" indicator orders
206  hv   208       LPC |
207  zq                 |
208  hv   210       LQC |
209  zq                 |
210  hv   212       LRC )
211  zq
212  arn  211       IZB )
213  ar   211       IZA |
214  zq             LZB |
215  zq             LZA |
216  pmn            IZC |
217  zq             NZB |
218  zq             NZA  > proving Z combinations
219  pin                |
220  qq             IZB |
221  ar   212           |
222  zq             NZB |
223  pin                |
224  qq             IZA |
225  ar   213           |
226  zq             NZA )
```

```
227   pi
228   hv   198     LKA   loop H, test of Z
229   hv   41
```

[Tromletest 4. This program writes random numbers on previous selected drum tracks, reads and compares the contents word by word incl. flag bits. Calculations may be performed simultaneously with the drum test by means of an adder test.

After the message ,klar, the typewriter is ready for input of 3 arbitrary numbers (each terminated by CR) intended for the production of the random test numbers.

The program has several adjustment variations. The p-register contains the track no. to be tested. The five last bits of the in-register have the following meaning:

```
  KB=0 :  Adder test simultaneously with the drum test.
  KB=1 :  Either adder test or drum test depending on when KB is set equal to 1.
  KA=0 :  No stop if adder error.
  KA=1 :  Stop if adder error.
          The typewriter writes ,1, if adder test is ok, and ,3, if adder error.
  RB=0 :  No counting in the p-register. Test of the same track.
  RB=1 :  Counting in the p-register. Track no. is increased with 1.
  RA=0 :  Output of number of errors per track.
  RA=1 :  Output of number of errors totally only.
  QB=0 :  Output of the read and written bit pattern, track no. and word no.
          if error.
  QB=1 :  No output of the read and written bit pattern, track no. and word no.
```

By entry a normal adjustment is set intended for a routine test of tracks 1-319 and output only of the number of errors totally (cf. the instructions in cell 42). By installations with Three Drum Cabinet the instruction in cell 51 should be changed to: arn 959 D.

The only way to change the normal adjustment is to change the contents of the in- and p-registers manually and jump to cell 43. The program starts with the instruction ,zq 0 LKB, and stops only if the operator interferes.]

```
i= 41
[  41] zq            LKB        [ Stop if KB=1]
[  42] pi    7     ,pp          [ Normal adjustment. Start on track 1]
[  43] gp    53    ,hs    89    [ Jump to input of 3 numbers]
[  44] pt    61            t45  [ Initialising return jump to drum test]
[  45] hv    257   ,grn   76    [ Start adder test, clear error counter tot.]
[  46] pp    p1       LRB       [ Add 1 to track no.]
[  47] hs    187                [ Store inputed numbers]
[  48] hv    155   ,hv    191   [ Fill track, store numbers]
[  49] grn   77    ,hs    187   [ Clear error counter per track, store numbers]
[  50] hv    164   ,hv    199   [ Read track, transfer numbers]
[  51] arn   319   D            [ If more tracks]
[  52] ncp         ,hv    46    [ then jump to new test]
[  53] pp          ,sy    64    [ else reset start track, write CR]
[  54] arn   76    ,hs    203   [ Jump to output of number of errors totally]
[  55] sy    64    ,hh    45    [ Write CR, jump to new test]
[  56] pt    61           V163  [ Entry interrupt drum test]
[  57] pt    61           t168  [ Entry interrupt drum test]
[  58] pm    195   ,arn   196   [ Reset R and M before]
[  59] gs    61    ,hv    290   [ jump to adder test]
[  60] gm    195   ,gr    196   [ Entry interrupt adder test]
[  61] ps          ,hh          [ Jump to drum test]
[  62] qq
[  63] qq                           [ Working cells]
[  64] qq
[  65] qq
[  66] qq
[  67] qq
[  68] qq
[  69] qq
[  70] qq
[  71] 0/1023/1023/1023             [ Constant]
[  72]       1                      [ Constant]
[  73] qq
[  74] qq
[  75] qq
[  76] qq
[  77] qq


                                    [ Subroutine, produce random test number]
[  78] pm   68                      [ From the numbers in cells 66, 67, and 68]
[  79] arn  68    X                 [ 3 numbers are produced to be placed in the]
[  80] mk   66                      [ same cells. The contents of R is on exit]
[  81] sc   67                      [ depending on the contents of these cells]
[  82] dl   68    X                 [ and used as a test number. Possibly]
[  83] ac   66        IOA           [ overflow is indicated in respective OA]
[  84] ml   67                      [ and OB for later check of flag bits.]
[  85] ac   68        IOB
[  86] hr   s1


                                    [ Subroutine, input of 3 numbers]
[  87]  0                           [ Working cell]
[  88]  10                          [ Constant]
[  89] pa   106             t62     [ Reset start cell]
[  90] pa   107             t2      [ Reset counting]
[  91] vy   17    ,sy   64          [ Select typewriter as I/O, write CR]
[  92] sy   34    ,sy   35          [ Write k, write l]
[  93] sy   49    ,sy   41          [ write a, write r]
[  94] sy   64                      [ Write CR]
[  95] pt   102             t106    [ Set jump address for positive number]
[  96] xrn        ,lyn  87          [ Clear M, input of 1 character]
[  97] nc   32    ,hv   100         [ If character ≠32 then jump]
[  98] pt  102              t105    [ else set jump address for negative number]
[  99] hh   96                      [ Jump to input of next character]
[ 100] ca   16    ,hvn  103         [ If character=0 then clear R and jump]
[ 101] ca   64                      [ If character=64]
[ 102] xr         ,hv               [ then exchange M and R and jump to storing]
[ 103] tk   -30   ,ml   88          [ else produce a decimal number in M]
[ 104] hh   96                      [ Jump to input of next character]
[ 105] mt   -1    D                 [ If negative number then change sign]
[ 106] gr   62              t1      [ Store the decimal number in working cell]
[ 107] bt   2               t-1     [ Counter for input of 3 numbers]
[ 108] hv   95                      [ Jump within 3 numbers]
[ 109] hr   s1                      [ else exit]


                                    [ Subroutine, output of bit pattern from R]
[ 110] pa   121             t3      [ Prepare output in 4 groups]
[ 111] tl   -10                     [ Prepare output of 1 bit at a time]
[ 112] pa   118             t9      [ Prepare 10 positions in each group]
[ 113] mb   71                      [ Clear R pos. 0-9]
[ 114] tl   1                     [ Produce 1 bit]
[ 115] ca                         [ If it is 0]
[ 116] sy   16    V               [ then write 0]
[ 117] sy   1                     [ else write 1]
[ 118] bt   9              t-1    [ Counter for output of 10 pos.]
[ 119] hv   113                   [ Jump back within a group]
[ 120] sy                         [ else write SP]
[ 121] bt   3              t-1    [ Counter for output of 4 groups]
[ 122] hv   112                   [ Jump back within 4 groups]
[ 123] sy   1     V  LPA          [ else write ,1, or ,0, according to PA]
[ 124] sy   16                    [ which corresponds to bit 40]
[ 125] sy   1     V  LPB          [ Write ,1, or ,0, according to PB]
[ 126] sy   16                    [ which corresponds to bit 41]
[ 127] hr   s1                    [ Exit]


                                  [ Subroutine, output of track no. and word no.]
[ 128] sy         ,sy             [ Write SP, write SP]
[ 129] sy   34    ,sy   49        [ Write k, write a]
[ 130] sy   37    ,sy   49        [ write n, write a]
[ 131] sy   35    ,sy             [ write l, write SP]
[ 132] arn  p     D               [ Prepare output]
[ 133] ck         ,tl   -30       [ of track no.]
[ 134] hs   203                   [ Jump to output of track no.]
[ 135] sy         ,sy             [ Write SP, write SP]
[ 136] sy   38    ,sy   41        [ Write o, write r]
[ 137] sy   52    ,sy   37        [ write d, write n]
[ 138] sy   41    ,sy   59        [ write r, write .]
[ 139] srn  314   D               [ Prepare output]
[ 140] ar   170   ,tl   -30       [ of word no.]
[ 141] hs   203                   [ Jump to output of word no.]
[ 142] hr   s1                    [ Jump back to error output]


                                  [ Subroutine, error output]
[ 143] arn  72    ,ac   76        [ Add 1 to number of errors totally]
[ 144] ac   77                    [ Add 1 to number of errors per track]
[ 145] hv   180     LQB           [ If QB=1 then jump back without output]
[ 146] arn  69      IPC           [ else prepare output of test number]
[ 147] vy   16    ,sy    64       [ Select typewriter, write CR]
[ 148] sy   64                    [ Write CR]
[ 149] hs   110                   [ Jump to output of bit pattern]
[ 150] sy   64                    [ Write CR]
[ 151] arn (170)    IPC           [ Prepare output of read test number]
[ 152] hs   110                   [ Jump to output of bit pattern]
[ 153] hs   128                   [ Jump to output of track no. and word no.]
[ 154] hv   180                   [ Jump back to next test]


                                  [ Fill drum track]
[ 155] pa   158            t215   [ Reset start address]
[ 156] pa   159            t39    [ Reset counting]
[ 157] hs   78                    [ Jump to produce random test number]
[ 158] gr   215     MOC t1        [ Store result incl. flag bits]
[ 159] bt   39             t-1    [ Counter for 40 words]
[ 160] hv   157                   [ Jump back within a track]
[ 161] vk   p     ,sk   216       [ else select actual track no., write on track]
[ 162] hv   56      NKB           [ Interrupt if KB=0]
[ 163] vk   p     ,hh   48        [ Exit]
```

```
                                          [ Read track and check]
[ 164] pa   180            t39            [ Reset counting]
[ 165] pa   170            t313           [ Reset start address]
[ 166] vk   p     ,lk   314               [ Select actual track no., read track]
[ 167] hv   57      NKB                   [ Interrupt if KB=0]
[ 168] vk   p     ,hs   78                [ Jump to produce random test number]
[ 169] gr   69      MOC                   [ Store test number incl. flag bits]
[ 170] sr   313            t1             [ Subtract read number]
[ 171] hv   143     NZ                    [ Jump to error output if R pos. 0-39≠0]
[ 172] hv   175     LOB
[ 173] hv   143     LB                    [ Jump to error output if OB=0∧b=1]
[ 174] hv   176
[ 175] hv   143     NB                    [ Jump to error output if OB=1∧b=0]
[ 176] hv   179     LOA
[ 177] hv   143     LA                    [ Jump to error output if OA=0∧a=1]
[ 178] hv   180
[ 179] hv   143     NA                    [ Jump to error output if OA=1∧a=0]
[ 180] bt   39             t-1            [ Counter for 40 words]
[ 181] hv   168                           [ Jump back within a track]
[ 182] hh   50      LRA                   [ else if RA=1 then return jump]
[ 183] vy   16    ,sy   64                [ else select typewriter, write CR]
[ 184] arn   77                           [ Prepare output of number of errors per track]
[ 185] hs   203                           [ Jump to output of number of errors]
[ 186] sy   64    ,hh   50                [ Write CR, exit]


                                          [ Transfers of working cells]
[ 187] arn  63    ,gr   66                [ Storing of the 3 inputed numbers]
[ 188] arn  64    ,gr   67
[ 189] arn  65    ,gr   68
[ 190] hr   s1
[ 191] arn  66    ,gr   73                [ Storing of the 3 numbers produced by]
[ 192] arn  67    ,gr   74                [ passage of Produce random test number]
[ 193] arn  68    ,gr   75
[ 194] hv   49
[ 195] qq                                 [ Working cells]
[ 196] qq
[ 197] qq
[ 198] qq
[ 199] arn  73    ,gr   63                [ Results for later use as start numbers]
[ 200] arn  74    ,gr   64
[ 201] arn  75    ,gr   65
[ 202] hv   51


                                          [ Output of number of errors]
[ 203] pa   210            t4             [ Adjustment for 5 digits]
[ 204] dk   215                           [ Transform to machine number]
[ 205] ar   214   X                       [ Possibly round off, exchange M and R]
[ 206] mln  213                           [ Prepare ]
[ 207] tk   30    ,ga   209               [ output of 1 digit]
[ 208] sy   16    V  LZ                   [ If digit=0 then write 0]
[ 209] sy                                 [ else write digit]
[ 210] bt   4              t-1            [ Counter for output of 5 digits]
[ 211] hv   206                           [ Jump within 5 digits]
[ 212] hr   s1                            [ else exit]


[ 213]  10                                [ Constants]
[ 214]   1
[ 215]  100000
[ 216] qq                                 [ 40 cells for the instruction sk 216]
```

```
i= 256                                [ Simultaneous calculating by adder test 5]
[ 256] gr   309                       [ Only intended for corrections]
[ 257] arn  304    ,vy    16          [ Entry by start of adder test]
[ 258] gr             M               [ Set address for floating-point overflow]
[ 259] pan  296    X      t1
[ 260] arfn 305
[ 261] grf  310
[ 262] arfn 306
[ 263] grf  311
[ 264] arfn 307
[ 265] grf  312                       [ Transfers of test numbers]
[ 266] arfn 308                       [ Adder test]
[ 267] grf  313
[ 268] arf  310
[ 269] mkf  311
[ 270] acf  312
[ 271] anf  313
[ 272] mlf  310    X
[ 273] scf  311
[ 274] srf  312
[ 275] mtf  313
[ 276] grf  310
[ 277] snf  311
[ 278] dkf  312
[ 279] gmf  313
[ 280] dlf  310
[ 281] ar   310
[ 282] mk   311
[ 283] ac   312
[ 284] an   313
[ 285] ml   310    X
[ 286] sc   311
[ 287] sr   312
[ 288] tl   -1
[ 289] hk   60        NKB             [ Jump to drum test if transfer is finished]
[ 290] mt   313
[ 291] gr   310
[ 292] sn   311
[ 293] dk   312
[ 294] gm   313
[ 295] dl   310
[ 296] bt   1             t1          [ Counter for 512 passages]
[ 297] hv   268                       [ Jump back within 512 passages]
[ 298] sr   309    ,ck                [ else subtract result]
[ 299] sy   1         LZ              [ Write ,1, if adder is ok]
[ 300] hv   259       LZ              [ Jump to new test if adder is ok]
[ 301] zq             LKA             [ Stop if adder error∧KA=1]
[ 302] sy   3                         [ Write ,3, if adder error]
[ 303] hv   259                       [ Jump to new test]
[ 304] hv   269
[ 305]  1234567890                    [ Test number]
[ 306]  2345678901
[ 307]  3456789012
[ 308]  4567890123
[ 309]  275/446/57/624                [ Result of adder test]
[ 310]  0
[ 311]  0
[ 312]  0
[ 313]  0
[ 314]  0                             [ 40 cells for the instruction lk 314]

e 41
```

[Test Program No. 14. Reader Test. 15/1-1965.

The program is to be used with an infinite 8-hole punched paper tape containing sets of characters with odd parity and values ranging from 0-127. Between each set of 128 characters there is about 1 inch of blank tape. Such tapes can be produced with Test Program A 206.

Input of the test tape should be started with blank tape and the program stops in case of error. When the START button is pressed, the program proceeds with the next character.

In case of error the character read is to be found in the R register and the character which it ought to have been in the p register. The p register can be altered manually during error stop.]

```
i=41
zq    LKB                ; stop if KB
zq    NKB                ; stop if not KB
vy                       ; choose reader
pm   127   D             ; set mask for bits 3-9
pp   1023                ; set first character to -1
lyn  r-4,     pp p+1     ; read, add 1 to p
cm   p     D             ; compare
zq                       ; error stop
hv   r-3                 ; jump
e41
```

Should it be impossible to input the test program by means of the reader, it can be done manually from the console by insertion of the following in cells 0-3:

```
[0]  lyn  2,    pp p+1; bits No. 8,19-22,24,25,31-35,38-40
[1]  cm p D           ; bits No. 20,24,25,28,29,32
[2]  zq               ; bit No. 25
[3]  hv               ; bits No. 20-22
```

Also the following must be inserted in the registers:

by = 0
M = $127\times_{10}$-9; bits No. 3-9
p = 1023 ; bits No. 0-9

The test is started by a jump to cell No. 0 and the test is performed as described above.

TEST B-25
GIER-DISK
RIALTO 3/6/66
JFM

PROGRAM FOR TEST OF THE DISKFILE

GENERAL

The program occupies cells 41-501 of the core store, and use cells 640-959 and 980-1199 during the run. The program is published in a condensed version (a B-tape) to be read by the track 0 readprogram. The program starts in cell 41 with the instruction zq LKB.

FUNCTIONS

After reading in some questions are typed on the console typewriter. The answers will define the run-path of the program. It is possible to select between writing with stop after writing or reading only. In this way one read and test a diskkit written days ago. If KA is set to 1 during the read cycle the reading will continue until KA is set to zero again. One can select whether one want to write the tracknumbers (by tracknumbers I shall understand the AneLex tracknumbers in the range from 0 to 99) on the tracks, or one can write and test with a bitpattern inserted in the R-register positions 0-41. After a writecycle with bitpattern, the pattern will be typed out (use paper in A4-horizontal format). The information (tracknumbers or bitpattern) will be written in all words of all tracks on the file. There is two different ways of selecting tracks for writing or reading. The first one is called the normal way. In this the tracknumbers are increased by one from 0 to 99. The second, called the modular way, select tracks in the following manner, always starting with 0:

0-99-1-98-2- . . . -50

Using this method of trackselection the arms will be moved over several tracks.

USE OF KA AND KB

The setting of KA and KB will have the following effects:

KA=0	: Normal read-write cycle
KA=1	: After one read cycle the reading is repeated
KB=0	: Normal track selection during reading and writing
KB=1	: The first 32 sectors of track 0 are not tested (these sectors are normally locked for writing)

PRINTING OF ERRORS

When an error has been found during the check reading, an errormessage will be typed. It will be in one of the following two formats:

1) In the addressing mode:
```
w <written tracknumber><head number><sector number>      (black)
r <read tracknumber>    <tensgroup>                      (red  )
w <written flagbits>                                     (black)
r <read flagbits>                                        (red  )
```

2) In the bitpattern mode:
```
w <written bitpattern> <head number><sector number>      (black)
r <read bitpattern>    <tensgroup>  <tracknumber>        (red  )
w <written flagbits>                                     (black)
r <read flagbits>                                        (red  )
```

If no error has been detected during the read cycle the message <no errors> will be typed.

Jan Flemming Madsen

```
[Test Program No. A 206.
Production of Tape for Reader Test No. 14. 15/1-1965.
-----------------------------------------------------

Punches characters with the values 0-127 and stops. When the
START button is pushed, the program is repeated. ]

i = 10
vy  32                             ; choose punch
pa  r1     t 1023                  ; set first character to -1
sy         t 1                     ; print character  +1
bs  (r-1)  t 126                   ; next order if 127
zq                                 ; stop, jump to first character
hv  r-3                            ; jump to next character
e10
```

TEST A-209
Diskfile on Buffer
RIALTO 21/6/66
JFM

Addressing test for diskfile connected to the buffer store.

The program writes the tracknumbers in position 39 of all words in all blocks on the file. The blocklength is normally 400 words (i.e. max. blocklength), but can be changed by a redefinition during reading of the programtape. After reading of a few lines the reading stops and one can type:

```
d d30 = <wanted blocklength>
I
```

After reading in and typing of filenumber etc. the program starts. During run KA=0 will give rewriting after one checkcycle, and KA=1 will give rereading.

PRINTING OF ERRORS

If an error is detected the following line will be typed, one line per word with error:

```
+ <correct tracknumber>-<tracknumber from file> w <wordnumber in block>
```

Jan Flemming Madsen

TEST A-210
Diskfile on Buffer
RIALTO 21/6/66
JFM

TB-TEST OF DISKFILE ON BUFFER

The program writes specified bitpatterns on the file and checks the written information. The area to be tested is specified by the user.

The program occupies cells 41-281 of the core store and use cells 622-1022 during running. The first instruction to be executed after reading in the program is zq LKB.

RUNNING

After start the program ask for the filenumber (0, 1, 2 or 3). Then the users types the wanted blocklength (in the range from 0 to 400). At last the starttrack and endtrack is typed. The condition for acceptance of the typed values for these are:

$$0 < \text{first track} \leq \text{last track} \leq 1199$$

After this the wanted bitpattern is inserted in the R-register positions 0-41 and the NORMAL START button is depressed. Now the bitpattern given will be written on the given area and checked. After checking, the reading will continue if KA=1. If KA=0 and KB=1 the program will rewrite the same area. If both KA and KB is zero the program jump to the startsituation.

ERRORSTOP

When errors is detected the machine stop with the r1-register set to either 162 or 165.

1) r1=162, INFORMATION ERROR
   The correct bitpattern is in the M-register,
   the wrong in the R-register.
2) r1=165, ERROR IN FLAGBITS
   The correct marks are in R position 8 and 9,
   the wrong marks are in the In-register positions RA and RB.

If a parity error is found the program tries to read the track in question two times more. If this still fails a <p> is typed and the test continue with the next track.

Jan Flemming Madsen

TEST A-211
Diskfile on Buffer
RIALTO 21/6/66
JFM

FAST INFORMATIONTEST OF DISKFILE ON BUFFER

The program writes some, more or less, random numbers on the entire file and check this information. The check is performed as a sumcheck, and the program types 'sum-error' in red whenever this summation fails. At the startpoint the filenumber (0, 1, 2 or 3) should be typed.

Jan Flemming Madsen

TEST 212a
GIER-DISK
RIALTO 20/6/66
JFM

TEST OF THE ARM MOVING

The program occupies cells 41-63 of the core store.

METHOD

The program READS the entire diskkit sector by sector. The sectors to be read are selected in the following manner:

1.	track	0,	sector	0
2.	-	99,	-	959
3.	-	0,	-	1

etc. up to:

9599. -	99,	-	959
9600. -	0,	-	0

After reading the program select sector 0 of the SLIP system and jump to the first instruction. Depending on the information on the given diskkit, the program will end with one of the messages:

hp-knap xx
algol
SUM-ALGOL
FEJL

Jan Flemming Madsen

Beschreibung vom Programm Roots, das beim Übernahmetest von GIER angewendet wird.

Das Programm testet die Prozedur Roots, die sämtliche Komplexwurzeln in einem n'ten Grad Polynom Pn (z) mit Komplexkoeffizienten findet.
Das geschieht, indem man Nullstellen für $F(x,y) = |Fn(x + iy)|^2$ mit Hilfe einer Gradientmethode sucht.

Das Testprogramm, das ein ALGOL Programm ist, wird eingelesen und übersetzt, darauf werden die n (n=75) komplexen Zahlen, die nach dem Programm auf demselben Streifen stehen, eingelesen.

Das Programm generiert nun die Koeffizienten in diesem Polynom, welches diese Zahlen als Wurzel hat und wendet diese Koeffizienten als Parameter für Roots an. Die Lösung kommt als Ausgabe auf den Lochstreifenlocher.

Danach beginnt automatisch das Aufsuchen der Wurzeln im gleichen Polynom, und die Wurzeln von jedem Durchlauf können nun verglichen werden. Ein Durchlauf dauert etwa 40 Minuten.

Will man untersuchen, wie weit die Maschine in einem Durchlauf gekommen ist, drückt man K B und jedesmal wenn die Maschine eine Wurzel gefunden hat, wird sie - solange K B gedrückt ist - schreiben, wieviele Wurzeln noch zu finden sind.

Ved ny hjelp (Hjelp 3) skal strimlen startes efter the header

START

Tid / celle	Før start	1. ord indlæst og lagret	2. ord indlæst og lagret	3. ord indlæst og lagret	4. ord indlæst og lagret
34	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA
35	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1
36	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508
37	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB
38	hv 35 LZB	hv 35 LZB	hv 35 LZB	hv 35 LZB	hv 35 LZB
39	gr 41 MRC t-1	gr 40 MRC t-1	gr 42 MRC t-1	gr 41 MRC t-1	gr 40 MRC t-1
40		hv 34	hv 34	hv 34	gr 46 MRC t-2
41				hv 34	hv 34
42					
43					
44					
45					
46					

Tid / celle	4. ord indlæst og lagret	5. ord indlæst og lagret	5. ord indlæst og lagret	6. ord indlæst og lagret	Sidste ord indlæst og lagret
34	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	
35	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	
36	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	
37	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	
38	hv 35 LZB	hv 35 LZB	hv 35 LZB	hv 35 LZB	
39	gr 40 MRC t-1	tl 3	tl 3	tl 3	
40	gr 44 MRC t-2	gr 44 MRC t-2	gr 42 MRC t-2	gr 1 MRC t-1	gr 41 MRC t-1
41	hv 34	hv 34	hv 34	hv 34	↑
42			tl 3	tl 3	
43					
44	gr 46 MRC t-2	gr 46 MRC t-2	gr 46 MRC t-2	gr 46 MRC t-2	
45					Indlæste program
46					↓

REGNECENTRALEN Dansk Institut for Matematikmaskiner	Tegnet	10.8.63.
	Kontrol	
	Godk.	

Startprocedure ved indlæsning med kanal 0

Gl. Hjelp

DECIMAL	OPE.	BINARY VALUE						OPE.	DECIMAL	BINARY VALUE					
		20	21	22	23	24	25			20	21	22	23	24	25
0	qq	.	.	.	.	.	.	ab	9	.	.	1	.	.	1
1	zq	.	.	.	.	.	1	ac	6	.	.	.	1	1	.
2	ar	.	.	.	.	1	.	an	4	.	.	.	1	.	.
3	sr	.	.	.	.	1	1	ar	2	.	.	.	.	1	.
4	an	.	.	.	1	.	.	bs	49	1	1	.	.	.	1
5	sn	.	.	.	1	.	1	bt	39	1	.	.	1	1	1
6	ac	.	.	.	1	1	.	ca	25	.	1	1	.	.	1
7	sc	.	.	.	1	1	1	ck	19	.	1	.	.	1	1
8	mb	.	.	1	.	.	.	cl	20	.	1	.	1	.	.
9	ab	.	.	1	.	.	1	cm	38	1	.	.	1	1	.
10	mt	.	.	1	.	1	.	dk	13	.	.	1	1	.	1
11	mk	.	.	1	.	1	1	dl	14	.	.	1	1	1	.
12	ml	.	.	1	1	.	.	ga	22	.	1	.	1	1	.
13	dk	.	.	1	1	.	1	gc	47	1	.	1	1	1	1
14	dl	.	.	1	1	1	.	gg	46	1	.	1	1	1	.
15	nk	.	.	1	1	1	1	gi	29	.	1	1	1	.	1
16	nl	.	1	.	.	.	.	gk	54	1	1	.	1	1	.
17	hr	.	1	.	.	.	1	gm	26	.	1	1	.	1	.
18	tl	.	1	.	.	1	.	gp	42	1	.	1	.	1	.
19	ck	.	1	.	.	1	1	gr	21	.	1	.	1	.	1
20	cl	.	1	.	1	.	.	gs	61	1	1	1	1	.	1
21	gr	.	1	.	1	.	1	gt	23	.	1	.	1	1	1
22	ga	.	1	.	1	1	.	hh	60	1	1	1	1	.	.
23	gt	.	1	.	1	1	1	hk	34	1	.	.	.	1	.
24	tk	.	1	1	.	.	.	hr	17	.	1	.	.	.	1
25	ca	.	1	1	.	.	1	hs	50	1	1	.	.	1	.
26	gm	.	1	1	.	1	.	hv	56	1	1	1	.	.	.
27	pm	.	1	1	.	1	1	il	44	1	.	1	1	.	.
28	xr	.	1	1	1	.	.	is	36	1	.	.	1	.	.
29	gi	.	1	1	1	.	1	it	37	1	.	.	1	.	1
30	ps	.	1	1	1	1	.	lk	52	1	1	.	1	.	.
31	pp	.	1	1	1	1	1	ly	59	1	1	1	.	1	1
32	pa	1	.	.	.	.	.	mb	8	.	.	1	.	.	.
33	pt	1	.	.	.	.	1	mk	11	.	.	1	.	1	1
34	hk	1	.	.	.	1	.	ml	12	.	.	1	1	.	.
35	pi	1	.	.	.	1	1	mt	10	.	.	1	.	1	.
36	is	1	.	.	1	.	.	nc	43	1	.	1	.	1	1
37	it	1	.	.	1	.	1	nk	15	.	.	1	1	1	1
38	cm	1	.	.	1	1	.	nl	16	.	1	.	.	.	.
39	bt	1	.	.	1	1	1	ns	40	1	.	1	.	.	.
40	ns	1	.	1	.	.	.	nt	41	1	.	1	.	.	1
41	nt	1	.	1	.	.	1	pa	32	1	.	.	.	.	.
42	gp	1	.	1	.	1	.	pc	48	1	1	.	.	.	.
43	nc	1	.	1	.	1	1	pi	35	1	.	.	.	1	1
44	il	1	.	1	1	.	.	pm	27	.	1	1	.	1	1
45	us	1	.	1	1	.	1	pp	31	.	1	1	1	1	1
46	gg	1	.	1	1	1	.	ps	30	.	1	1	1	1	.
47	gc	1	.	1	1	1	1	pt	33	1	.	.	.	.	1
48	pc	1	1	.	.	.	.	qq	0	.	.	.	.	.	.
49	bs	1	1	.	.	.	1	sc	7	.	.	.	1	1	1
50	hs	1	1	.	.	1	.	sk	53	1	1	.	1	.	1
51	vy	1	1	.	.	1	1	sn	5	.	.	.	1	.	1
52	lk	1	1	.	1	.	.	sr	3	.	.	.	.	1	1
53	sk	1	1	.	1	.	1	sy	58	1	1	1	.	1	.
54	gk	1	1	.	1	1	.	tk	24	.	1	1	.	.	.
55	vk	1	1	.	1	1	1	tl	18	.	1	.	.	1	.
56	hv	1	1	1	.	.	.	ud	63	1	1	1	1	1	1
57	zj	1	1	1	.	.	1	us	45	1	.	1	1	.	1
58	sy	1	1	1	.	1	.	vk	55	1	1	.	1	1	1
59	ly	1	1	1	.	1	1	vy	51	1	1	.	.	1	1
60	hh	1	1	1	1	.	.	xr	28	.	1	1	1	.	.
61	gs	1	1	1	1	.	1	zj	57	1	1	1	.	.	1
62	zl	1	1	1	1	1	.	zl	62	1	1	1	1	1	.
63	ud	1	1	1	1	1	1	zq	1	.	.	.	.	.	1

apr 67
jfm

TABLE OF INDICATORPARTS
(inactive are marked with an ×)

33	34	35	36	37	38	39		33	34	35	36	37	38	39	
.	.	.	.	.	.	.	I ×	1	.	.	.	.	.	.	N ×
.	.	.	.	.	.	1	IB ×	1	.	.	.	.	.	1	NB
.	.	.	.	.	1	.	IA ×	1	.	.	.	.	1	.	NA
.	.	.	.	.	1	1	IC ×	1	.	.	.	.	1	1	NC
.	.	.	.	1	.	.	IK	1	.	.	.	1	.	.	NK ×
.	.	.	.	1	.	1	IKB	1	.	.	.	1	.	1	NKB
.	.	.	.	1	1	.	IKA	1	.	.	.	1	1	.	NKA
.	.	.	.	1	1	1	IKC	1	.	.	.	1	1	1	NKC
.	.	.	1	.	.	.	IZ ×	1	.	.	1	.	.	.	NZ
.	.	.	1	.	.	1	IZB	1	.	.	1	.	.	1	NZB
.	.	.	1	.	1	.	IZA	1	.	.	1	.	1	.	NZA
.	.	.	1	.	1	1	IZC	1	.	.	1	.	1	1	NZC
.	.	.	1	1	.	.	IO ×	1	.	.	1	1	.	.	NO
.	.	.	1	1	.	1	IOB	1	.	.	1	1	.	1	NOB
.	.	.	1	1	1	.	IOA	1	.	.	1	1	1	.	NOA
.	.	.	1	1	1	1	IOC	1	.	.	1	1	1	1	NOC
.	.	1	.	.	.	.	IT ×	1	.	1	.	.	.	.	NT
.	.	1	.	.	.	1	ITB	1	.	1	.	.	.	1	NTB
.	.	1	.	.	1	.	ITA	1	.	1	.	.	1	.	NTA
.	.	1	.	.	1	1	ITC	1	.	1	.	.	1	1	NTC
.	.	1	.	1	.	.	IP ×	1	.	1	.	1	.	.	NP ×
.	.	1	.	1	.	1	IPB	1	.	1	.	1	.	1	NPB
.	.	1	.	1	1	.	IPA	1	.	1	.	1	1	.	NPA
.	.	1	.	1	1	1	IPC	1	.	1	.	1	1	1	NPC
.	.	1	1	.	.	.	IQ ×	1	.	1	1	.	.	.	NQ ×
.	.	1	1	.	.	1	IQB	1	.	1	1	.	.	1	NQB
.	.	1	1	.	1	.	IQA	1	.	1	1	.	1	.	NQA
.	.	1	1	.	1	1	IQC	1	.	1	1	.	1	1	NQC
.	.	1	1	1	.	.	IR ×	1	.	1	1	1	.	.	NR ×
.	.	1	1	1	.	1	IRB	1	.	1	1	1	.	1	NRB
.	.	1	1	1	1	.	IRA	1	.	1	1	1	1	.	NRA
.	.	1	1	1	1	1	IRC	1	.	1	1	1	1	1	NRC
.	1	.	.	.	.	.	M	1	1	.	.	.	.	.	L ×
.	1	.	.	.	.	1	MB	1	1	.	.	.	.	1	LB
.	1	.	.	.	1	.	MA	1	1	.	.	.	1	.	LA
.	1	.	.	.	1	1	MC	1	1	.	.	.	1	1	LC
.	1	.	.	1	.	.	MK ×	1	1	.	.	1	.	.	LK ×
.	1	.	.	1	.	1	MKB ×	1	1	.	.	1	.	1	LKB
.	1	.	.	1	1	.	MKA ×	1	1	.	.	1	1	.	LKA
.	1	.	.	1	1	1	MKC ×	1	1	.	.	1	1	1	LKC
.	1	.	1	.	.	.	MZ ×	1	1	.	1	.	.	.	LZ
.	1	.	1	.	.	1	MZB ×	1	1	.	1	.	.	1	LZB
.	1	.	1	.	1	.	MZA ×	1	1	.	1	.	1	.	LZA
.	1	.	1	.	1	1	MZC ×	1	1	.	1	.	1	1	LZC
.	1	.	1	1	.	.	MO ×	1	1	.	1	1	.	.	LO
.	1	.	1	1	.	1	MOB	1	1	.	1	1	.	1	LOB
.	1	.	1	1	1	.	MOA	1	1	.	1	1	1	.	LOA
.	1	.	1	1	1	1	MOC	1	1	.	1	1	1	1	LOC
.	1	1	.	.	.	.	MT ×	1	1	1	.	.	.	.	LT
.	1	1	.	.	.	1	MTB	1	1	1	.	.	.	1	LTB
.	1	1	.	.	1	.	MTA	1	1	1	.	.	1	.	LTA
.	1	1	.	.	1	1	MTC	1	1	1	.	.	1	1	LTC
.	1	1	.	1	.	.	MP ×	1	1	1	.	1	.	.	LP ×
.	1	1	.	1	.	1	MPB	1	1	1	.	1	.	1	LPB
.	1	1	.	1	1	.	MPA	1	1	1	.	1	1	.	LPA
.	1	1	.	1	1	1	MPC	1	1	1	.	1	1	1	LPC
.	1	1	1	.	.	.	MQ ×	1	1	1	1	.	.	.	LQ ×
.	1	1	1	.	.	1	MQB	1	1	1	1	.	.	1	LQB
.	1	1	1	.	1	.	MQA	1	1	1	1	.	1	.	LQA
.	1	1	1	.	1	1	MQC	1	1	1	1	.	1	1	LQC
.	1	1	1	1	.	.	MR ×	1	1	1	1	1	.	.	LR ×
.	1	1	1	1	.	1	MRB	1	1	1	1	1	.	1	LRB
.	1	1	1	1	1	.	MRA	1	1	1	1	1	1	.	LRA
.	1	1	1	1	1	1	MRC	1	1	1	1	1	1	1	LRC

APR. 67
JFM

Testprograms Type C
RIALTO, 19.8.1968
JFM

The usual testprograms of B type can not be read by the primitive input program on track 0 in HELP 3. This paper describes how the modified tapes, called C programs, may be read by HELP 3.

FORMAT OF C TAPES

The paper tapes containing C programs consists of a short header followed by the corresponding B program. The header can be read by HELP 3 and enables the user to read the B program.

INPUT PROCEDURE

The header is followed by a tape gap 1" long. The reading may start before the header or in this gap. If the should be read by HJÆLP one starts in the gap, and if HELP 3 is used for reading one starts before the header. In the last case the input procedure is as follows:

(1) Read track 0 to the core store by
  a. Pressing RESET or MICROTEMPI STOP.
  b. Pressing HP-BUTTON while NORMAL STOP is pressed.

eller r< (2) Place the tape in the reader and start GIER in cell 23 by
  a. Setting r1=23
  b. BY=0
  c. Pressing NORMAL START

(3) When the program types
  t < number of program >
the header hes been read and the remaining part of the tape (the B program) will be read immidiately.

(4) When the B program has been read, the program in question starts running.

The actual procedure between (2) and (3) above is:

(A) A program (in bin 0 form) containing the primitive loader of HJÆLP is read to the core store in cells 493-512 and entered.

(B) This program types the test number and moves the loader to core 34-39.

(C) The program jumps to core 34 and from this point the action is as if track 0 of HJÆLP has been called.

PUNCHING OF C PROGRAMS

C programs are punched by the slip program A-224, "B to C Producer", and this program and how to use it is described in a seperate paper.

Rialto, 25. april 1968
JFM
t15

Test 15, Input and Output of 5 track paper tape

The program is used for test of input (from RC-2000) and output (to Facit Punch) of 5 track paper tape. The program is published as a binary paper tape, and may be loaded to the core store by the loaderprogram on track 0 for HJÆLP or HELP 3.

Subprograms

The program may be used in four different ways. The actual subprogram is selected when the program types:

Type c for copy, t for typeout, i for inputtest, o for outputtest:

The programs which may be selected arè:

1. Copying of 5 track paper tape	(type c)
2. Output of a special 5 track test tape	(type o)
3. Input and test of the tape produced by program 2	(type i)
4. Conversion from 5 track paper tape in TELEX code to typewriter output in GIER code	(type t)

The subprograms 1,2 and 4 may be stopped by setting KA=1. This causes a return to the startsituation and a new subprogram may be selected.

Test C-16

GIER-CP

Technical Center, 26.9.1968
JFM

The program is used for test of Mode 5 in GIER. Mode 5 is executed whenever the HP-button is pressed. The program is published as a B-C program only.

Method

After start (in Core[41]) the following happens:

1. Track 0 is saved in the core store during the test. Core[128:167]:=Track[0];
2. A special track 0 (part of the program) is written.
3. Track 38 is destroyed with special information.
4. The instructions

```
[512] vy      0, vk     87
[513] hv    512
```

are moved to Core[512:513] and entered.

The program will now loop in 512 and 513 until the HP-button is pressed. When this happens the following test sequence is executed:

1. Core[0:39] is stored on track 38 (by Mode 5)
2. Track 0 is read to the core store (Core[0:39]:=Track[0]). (by Mode 5)
3. The instruction counter CT (r1) is stored in the address part of Core[0] and the remaining bits cleared. (by Mode 5);
4. Core[1] is enterdd in either rigth halfword or in left halfword. The entry depends on the state of the h-bit when HP was pressed. (by Mode 5);
5. Now the functions 1-4 above are checked by the program on track 0.
6. A jump to the instructions in Core[512:513] is performed, and the test continue.

In case of errors, messages are given on the typewriter.

Use of KA,KB

KB=1 The program stops in Core[41] when the tape has been read.

KA=0 The normal state of KA during the test.

KA=1 After press on HP-button and the following test, the old track 0 is reestablished. One more press on HP will then activate the normal function of track 0 and the test is finished.

Error Messages

The following five messages are possible during the test, and the meaning is considered evident:

1. wrong tk stored
2. wrong r1 stored
3. -entry in Core[1]
4. Core[0:39] not stored on track 38
5. track 0 not transferred to core store

```
;slip<
;Test of HP function and mode 5, 1.8.68.4

b i=41, a40

[Core part of test,executed after track 0]

        zq              LKB         ; if kbon then STOP
        vy     512, grn      -1 ; set HP-inhibit; sum:=0;
        pa      a7   t      207 ; setsum of track 38:
        pa      a8   t       39 ;
a7:     arn    207   t        1 ;
        ar     1.8   DLA        ; include marks in sum
        ar     1.9   DLB        ;
        sc      -1              ; sum:=sum-R;
a8:     bt      39   t       -1 ;
        hv      a7              ;
        pm      -1,  gm     228 ;
        vk       0   lk     128 ; save present track 0 in core;
        sk     168,  vk      38 ; set new track zero;
        sk     208,  lk       0
        sk     288,  vk       0 ; noise to track 38 after setting core;
        pm      a9,  arn    a10 ;
        gm     512   MA         ; set fixed program in core;
        gr     513   M          ; program used for intterrupt by HP;
        hv     512              ; goto program;

a9:
[512]   vy       0,  vk      87 ; release HP-inhibit; set tk to 87;
a10:
[513]   hv     512              ; goto Core[512];


a1:     gr     a12              ; saveR:=R; error in tk:
        hs     a11              ; writetext(<<
        qq     a13              ; wrong tk stored>);
        arn    a12,  hr      s1 ; R:=saveR; return;

a2:     gr     a12              ; saveR:=R; error in OT:
        hs     a11              ; writetext(<<
        qq     a14              ; wrong r1 stored>);
        arn    a12,  hr      s1 ; R:=saveR; return;

a3:     hs     a11              ; improper entry to track 0: writetext(<<
        qq     a15              ; -entry in Core[1]>);
        hr      s1              ; return;

a6:     hs     a11              ; errror track 38: writetext(<<
        qq     a16              ; Core[0:39] not stored on track 38>);
        hr      s1              ; return;

a13:    k64 58twrong tk stored;58.
a14:    k64 58twrong r1 stored;58.
a15:    k64 58t-entry in Core[1];58.
a16:    k64 58tCore[0:39] not stored on track 38;58.
```

```
a11:  [textprint of HELP 3]
      gs     r5, an      s ;
      is     s1  MA        ;
      ps    (s), arn   r-2 ;
      pm      s. ps     s1 ;
      cl     -6; ca   10.5 ;
      ps     -1  hr     s1 ; return;
      ca   15.5, hh    r-4 ;
      tk     -4. ga     r2 ;
      ca     63, it      1 ;
      sy      0  hvn   r-5 ;

a12:  qq      0. qq      0 ; saveR;


d i=168

[New track 0  replacing the existing in the machine.
Handles and checks all actions taken by the HP-button
and mode 5]

[ 0]  qq      0   t    517 ; used for save OT;
      it    311, pt      0 ; normal entry: set bits(10-19) of Core[0];
      gk     -1, arn    -1 ; save by and tk; Rcount:=tk;
      vy    529  tk     10 ; set HP-inhibit;
      nc     87. hs     a1 ; if tk≠87 then call error;
      arn     0            ; R:=Core[0];
      ca    512, hv   ra17 ;
      ca    513  hv   ra17 ;
      ca      1, hv   ra17 ;
      ca      2, hv   ra17 ;
      ca      3. hv   ra17 ;
      hs     a2            ;
a17:  tk     10            ;
      nc    311, ca      0 ;
      hv   ra18            ;
      hs     a3            ;
a18:  vk     38, lk    288 ; read stored track 38 to core;
      vk     38, grn    -2 ; sum:=0;
      pa    ra4   t    287 ; setsum of track 38 in core[-2];
      pa    ra5   t     39 ;
a4:   arn   287   t      1 ;
      ar    1.8  DLA       ;
      ar    1.9  DLB       ;
[20]  sc     -2            ;
a5:   bt     39   t     -1 ;
      hv    ra4            ;
      arn    -2            ; R:=computed sum;
      hv   ra19   NZ       ; if R ≠ 0 then call error;
a20:
[27]  sk    208, lk      0 ; switch to new track 38;
[28]  hk     r1, hv      r ;
a19:  arn   289, nc    311 ;
      hs     a6            ;
      hv   ra20            ;
      qq                   ;
      qq                   ;
      qq                   ;
      qq                   ;
[34]  prn   1.3   DXIZA    ; Primitive input of HJELP:
      tl     -7, ly     r1 ;
      pi      0   IZA t508 ;
      xr      0   XIZB     ;
      hv     35   IZB      ;
[39]  gr     41   MRC  t-1 ;
```

d i=208
[Core 0:39 during the test. Stored on track 38 after HP.
The storing is performed by mode 5.]

```
[ 0]  qq                        ;
      vy    529, sy      64 ; writetext(<<
      sy     58, sy      19 ; track 0 not transferred to core store>);
      sy     41, sy      49 ;
      sy     51, sy      34 ;
      sy      0, sy      16 ;
      sy      0  sy      37 ;
      sy     38, sy      19 ;
      sy      0  sy      19 ;
      sy     41, sy      49 ;
[10]  sy     37, sy      18 ;
      sy     54  sy      53 ;
       sy     41, sy      41 ;
      sy     53  sy      52 ;
      hv     26             ;
      qq                    ;
      qq                    ;
      qq                    ;
      qq                    ;
      qq                    ;
[20]  qq                    ; sumbalance for track;
      qq                    ;
      qq                    ;
      qq                    ;
      qq                    ;
      qq                    ;
      qq                    ;
      qq                    ;
[28]  hk     r1, hv       r ; wait for transfer from drum;
      vk     38, sk     288 ; noise to track 38;
      hv    512   NKA       ; if - kaon then exit to core;
      vk      0  sk     128 ; regenerate track 0 of machine;
[30]  vk      0  hv     512 ; exit to core;
      qq                    ;
[34]  pmn   1.3   DXIZA     ; Primitive input of HJELP:
      tl     -7, lv      r1 ;
      pi      0   LZA t508  ;
      xr      0   XIZB      ;
      hv     35   LZB       ;
[39]  gr     41   MRC t-1   ;
```

s
e41

```
 25   1.8.68  849
start,image<
exit,10<
p26   1.8.68  e11
r<

1
294   41    248   300
a1     62
a2     66
a3     70
a4     188
a5     192
a6     73
a7     45
a8     49
a9     60
a10    61
a11    95
a12    105
a13    76
a14    79
a15    82
a16    87
a17    180
a18    184
a19    198
a20    196

 27   1.8.68  e42
binout,0,41..512<
print,p41..75,t76..94,p95..105,p168..247<
```

```
  41. zq                         LKB
  42. vy   512                 ,grn  -1
  43. ra   45                          t207
  44. ra   49                          t39
  45. arn  207                         t1
  46. ar   2                 D   LA
  47. ar   1                 D   LB
  48. sc   -1
  49. bt   39                          t-1
  50. hv   45
  51. pm   -1                 ,gm    228
  52. vk                      ,lk    128
  53. sk   168                ,vk    38
  54. sk   208                ,lk
  55. sk   288                ,vk
  56. pm   60                 ,arn   61
  57. gm   512                   MA
  58. gr   513                   M
  59. hv   512
  60. vy                      ,vk    87
  61. hv   512
  62. gr   105
  63. hs   95
  64. qq   76
  65. arn  105                ,hr    s1
  66. gr   105
  67. hs   95
  68. qq   79
  69. arn  105                ,hr    s1
  70. hs   95
  71. qq   82
  72. hr   s1
  73. hs   95
  74. qq   87
  75. hr   s1

  76.
wron
  77. g tk s
  78. tored
  79.
wron
  80. g r1 s
  81. tored
  82.
-ent
  83. ry in
  84. Core
  85. [1]
  86.
  87.
Co
  88. re[0
  89. :39
  90. ] not
  91.  store
  92. d on t
  93. rack 3
  94. 8

  95. gs   r5 [100]       ,an    s
  96. is   s1                NA
```

```
 98. pm   s                 ,ps   s1
 99. cl   -6                ,ca   160
100. ps   -1                ,hr   s1
101. ca   240               ,hh   r-4 [97]
102. tk   -4                ,ga   r2 [104]
103. ca   63                ,it   1
104. sy                     ,hvn  r-5 [99]
105. qq                     ,qq

168. qq                           t517
169. it   311               ,pt
170. gk   -1                ,arn  -1
171. vy   529               ,tk   10
172. nc   87                ,hs   62
173. arn
174. ca   512               ,hv   r6 [180]
175. ca   513               ,hv   r5 [180]
176. ca   1                 ,hv   r4 [180]
177. ca   2                 ,hv   r3 [180]
178. ca   3                 ,hv   r2 [180]
179. hs   66
180. tk   10
181. nc   311               ,ca
182. hv   r2 [184]
183. hs   70
184. vk   38                ,lk   288
185. vk   38                ,grn  -2
186. pa   r2 [188]                t287
187. pa   r5 [192]                t39
188. arn  287                     t1
189. ar   2                 D  LA
190. sr   1                 D  LB
191. sc   -2
192. bt   39                      t-1
193. hv   r-5 [188]
194. arn  -2
195. hv   r3 [198]             NZ
196. sk   208               ,lk
197. hk   r1 [198]          ,hv   r [197]
198. arn  289               ,nc   311
199. hs   73
200. hv   r-4 [196]
201. qq
202. qq
203. qq
204. qq
205. pmn  64                XD IZA
206. tl   -7                ,ly   r1 [207]
207. pi                        LZA t508
208. qq
209. vy   529               ,sy   64
210. sy   58                ,sy   19
211. sy   41                ,sy   49
212. sy   51                ,sy   34
213. sy                     ,sy   16
214. sy                     ,sy   37
215. sy   38                ,sy   19
216. sy                     ,sy   19
217. sy   41                ,sy   49
218. sy   37                ,sy   18
219. sy   54                ,sy   53
220. sy   41                ,sy   41
221. sy   53                ,sy   52
222. hv   26
223. qq
224. qq
225. qq
```

```
227. qq
228. qq
229. qq
230. qq
231. qq
232. qq
233. qq
234. qq
235. qq
236. hk   r1 [237]      ,hv   r [236]
237. vk   38            ,sk   288
238. hv   512             NKA
239. vk                 ,sk   128
240. vk                 ,hv   512
241. qq
242. pmn  64            XD LZA
243. tl   -7            ,ly   r1 [244]
244. pi                    LZA t508
245. xr                 X  LZB
246. hv   35               LZB
247. gr   41               MRC t-1
```

Rialto, 16 april 1968.
JFM
t 17

Test 17, Printer

The program is intended for test of the lineprinter. It is published as a binary papertape and may be loaded to the core store by the primitive input program on track 0 of either HJÆLP or HELP 3.

Method

The program prints characters in 159 positions on the printer.
The characters printed depends on the setting of KA and KB in the following manner:

1) KA=0, KB=0 : One line is printed consisting of one specific character. This character is typed on the typewriter when the program types Hej .

2) KA=1 , KB=0 : Each line contains all characters in both cases. The printing continues as long as KA is one. The first character in each line will be the one typed in the Hej-situation.

3) KA=0 , KB=1 : Each line contains the same character, but the both in upper case and lower case. The printerposition for shift between UC and LC will be shifted one towards left for each line. The first line will consist of the character typed in the Hej-situation.

4) KA=1 , KB=1 : Each line contain all characters and the printing is shifted one position towards left for each new line.

Blind symbols

All blind characters are printed as SPACE. This is also valid for STOPCODE and TABULATOR.

RC-RIALTO
21.3.1966
TESTPROGRAM B-18
JFM

The program is intended for test of disk files, in the cases where the disk unit replaces the usual drum unit; of course the program can be used for drum test, too.

PRINCIP
After reading in the program tape, the user specifies the area of the disk file he want to test. Then the program write known information in each word of this area. The information consist of word of the following format:

qq 'disknumber'. 9 + 'group'. 19 + 'sector'. 29 + 'wordnumber'. 39

After writing, this information is read to the core store in groups of eight sectors and compared with the correct information. This correct information is generated during reading and adjusted according to the current value of disk, group, sector, and word.

STORAGE LAYOUT

The core store is used as follows:

0 - 39	: track 0 of HELP
41	: the instruction zq LKB
42 - 143	: program administration
144 - 244	: number read routine
245 - 274	: number print routine
275 - 289	: text print routine
290 - 316	: routine for generation of correct information
317 - 334	: error print routine
335	: last group + 1
336	: last sector + 1
337	: disk unit number
338	: first group
339	: last group
340	: first sector
341	: last sector
342	: error used
343	: current value of group
344	: current value of sector
345	: address for choice of disk and tensgroup
346	: not used
347	: 4 (Floating)
348	: 10 -
349	: 961 -
350	: -40 (address)
351	: 960 -
352 - 382	: text strings
383 - 702	: 8 sectorsoof correct information
703 -1022	: 8 sectors read from the disk

RUNNING

After reading in the program types:

```
Disk No.:                   (type disk unit number)
First Group:                (type the wanted start group)
Last Group:                 ( -   -   -   end     -  )
First Sector:               ( -   -   -   start sector)
Last Sector:                ( -   -   -   end     -   )
```

First sector must be less than last sector and both < 960. First sector will be rounded to the nearest lower number divisible by 8. Last sector will be rounded to the nearest higher number divisible by 7, thus giving the possibility of writing and testing one entire track of eight sectors in one loop. First group must be less than or equal to last group and both less than 10. Disk unit must be less than 4. After typing last section the program starts. When one complete testcycle is finished, the program jumps to the start situation, if KB = 1 and stops in cell 41. If KB = 0, the program goes through another complete testcycle.

NUMBERS

The numbers to be typed must conform with the syntax for input in Gier-Algol III, because this is the input routine which is used.

ERRORS

When an error has been found, two lines are printed. One line in black showing the written information and one line in red showing the information which was read from the disk. The first error gives printing of a heading:

```
Disk Group Sect. Word
```

and the error print could be:

```
0      0    25     17        (black)
0      0   875     19        (red)
```

;   slip<

[Diagnostic program t 21 test of mode 1 and 4   oct.68   PEH

The program tests all possible combinations of the conditions in mode 1 and 4. If an error is detected output is given on typewriter according to the following table

output	type of non-successful function
g0, g1, g2	basic operation fullword, LH, RH halfword
a0, a1, a2	adress part of - - - - - -
n0, n1, n2	n-mark (S-mark) - - - - - -
i0, i1, i2	indirect adressing - - - - -
r0, r1, r2	relative-mark - - - - - -
s0, s1, s2	subroutinemark - - - - - -
p0, p1, p2	p-index - - - - - -
B	indicator condition
D	D-mark
f	floating-point mark
H	halfword mark added
h	- - - deleted
IK	IKC operation
K	K-part of indicator (bit pos.37)
T	T-part of indicator (bit pos.35)
Z	Z-part of indicator (bit pos.36)
L	LA or LB operation (bit pos.33, 34, 38, 39)
X	X-modification
V	V-modification
q	incremental operation
qs	incremental operation is erroneously treated as static or vice-versa

The letters may be combined, thus the message

isB

means that a mode 1 or 4 error is detected during execution of an instruction which is indirect adressed, s-marked and furnished with some indicator condition.

If KA is set during run the error message will be followed by a three-digit number which gives the adress of the cell from which the alarm was called.

Pressing of KB causes the machine to stop.

If no errors are detected an ok message is typed every 20 seconds.

Input of program.

The program may be read in like any other B or C test. However, if this does not work the program may be read in using as simple instructions as possible in this way:

1. RESET GIER.
2. place the tape in the reader starting in the middle of the 100 spaces found a few inches after the starting point used for old track 0.
3. clear the entire core store using hardware test 1.
4. clear R, M and by.
5. insert in cell 41 - 43 the following instructions:

41.	ly 43	bit pos. 4, 6,8, 9, 20, 21, 22, 24, 25
42	cl 76	- - 3, 6, 7, 21, 23
43.	gm	- - 21, 22, 24

6. start GIER in cell 41

When the program has been read in the primitive input program in cell 34 - 39 is re-established.   ]

```
        b a50, b15, c10

        i=27

[  27]      pm i+5     ;
[  28]      gm -2  M   ;
[  29]      pm i+4     ;
[  30]      gm 0   M   ;
[  31]      hv 40      ;
[  32]      ar a1      ;
[  33]      tk 42      ;
[  34]      pmn64XDIZA; primitive input
[  35]      tl -7,lyr1;
[  36]      pi LZAt508;
[  37]      xr X IZB   ;
[  38]      hv 35 LZB  ;
[  39]      gr41MRCt-1;
[  40]      tk 42      ;
[  41]      qq         ;
[  42]      vy 16      ;
[  43]      sy 29      ; red ribbon
[  44]      sy 58      ; LC
[  45]      sy 64      ;
[  46]      hv a0      ; goto start


;     alarm print

[  47]b0:   qq          ; work
[  48]      qq          ; constant
[  49]      sy 55       ; g
[  50]      qq          ;
[  51]      sy 49       ; a
[  52]      hv b1       ;
[  53]      sy 37       ; n
[  54]      hv b1       ;
[  55]      sy 57       ; i
[  56]      hv b1       ;
[  57]      sy 41       ; r
[  58]      hv b1       ;
[  59]      sy 18       ; s
[  60]      hv b1       ;
[  61]      sy 39       ; p
[  62]b1:   sy 16       ; [full-word]
[  63]      hv a2       ; goto s-print
[  64]      sy 55       ; g
[  65]      hv b2       ;
[  66]      sy 49       ; a
[  67]      hv b2       ;
[  68]      sy 37       ; n
[  69]      hv b2       ;
[  70]      sy 57       ; i
[  71]      hv b2       ;
[  72]      sy 41       ; r
[  73]      hv b2       ;
[  74]      sy 18       ; s
[  75]      hv b2       ;
[  76]      sy 39       ; p
[  77]b2:   sy  1       ; 1 [LH half-word]
[  78]      hv a2       ; goto s-print
[  79]      sy 55       ; g
[  80]       hv b3      ;
[  81]       sy 49      ; a
[  82]       hv b3      ;
[  83]       sy 37      ; n
[  84]       hv b3      ;
[  85]       sy 57      ; i
[  86]       hv b3      ;
[  87]       sy 41      ; r
[  88]       hv b3      ;
[  89]       sy 18      ; s
[  90]       hv b3      ;
[  91]       sy 39      ; p
[  92]b3:    sy  2      ; 2 [RH half-word]
[  93]       hv a2      ; goto s-print
[  94]       sy 60      ; B [indicator condition]
[  95]       sy 50      ;
[  96]       hv a2      ;
[  97]       sy 60      ; D
[  98]       sy 52      ;
[  99]       hv a2      ;
[ 100]       sy 54      ; f [floating-point mark]
[ 101]       hv a2      ;
[ 102]b4:    sy 60      ; H
[ 103]       sy 56      ; h
[ 104]       hv a2      ;
[ 105]       sy 60      ; IK [IKC operation]
[ 106]       sy 57      ;
[ 107]       sy 60      ; K
[ 108]       sy 34      ;
[ 109]       hv a2      ;
[ 110]       sy 60      ; L
[ 111]       sy 35      ;
[ 112]       hv a2      ;
[ 113]b5:    sy 40      ; q
[ 114]       hv a2      ;
[ 115]       sy 40      ; qs
[ 116]       sy 18      ; [static op treated as incremental
[ 117]       hv a2      ; or vice-versa]
[ 118]       sy 60      ; T
[ 119]       sy 19      ;
[ 120]       hv a2      ;
[ 121]       sy 60      ; X
[ 122]       sy 23      ;
[ 123]       hv a2      ;
[ 124]b6:    sy 60      ; XV
[ 125]       sy 23      ;
[ 126]       sy 60      ; V
[ 127]       sy 21      ;
[ 128]       hv a2      ;
[ 129]       sy 60      ; Z
[ 130]       sy 25      ;


;      s-print

[ 131]a2:    sy 58      ;
[ 132]       hv a3 NKA  ;
[ 133]       sy  0      ;
[ 134]       tk 42      ;
[ 135]       ar c2      ;
[ 136]       gr c3      ;
[ 137]       tl 80      ; R:=M:=0
[ 138]       gs i+1     ;
[ 139]       qq [s]     ;
[ 140]       ar i-1     ;
[ 141]       cl 10      ;
[ 142]       ml c       ;
[ 143]c4:    ck -10     ;
[ 144]       ca  0      ;
[ 145]       hv i+2     ;
[ 146]       hv i+3     ;
[ 147]       arn i+1    ;
[ 148]       qq 16      ;
[ 149]       ga i+1     ;
[ 150]       sy         ;
[ 151]       tk 42      ;
[ 152]       ar c3      ;
[ 153]       ar a1      ;
[ 154]       gr c3      ;
[ 155]       ca  0      ;
[ 156]       hv i+3     ;
[ 157]       ac c6      ;
[ 158]       hv a3      ;
[ 159]       ml c1      ;
[ 160]       hv c4      ;
[ 161]c0:    m 10-2     ;
[ 162]c1:    qq 10.39   ;
[ 163]c2:    qq 2.39    ;
[ 164]c3:    qq         ; work


;     message counter

[ 165]a3:    sy  0      ;
[ 166]       tk 42      ;
[ 167]       ar c6      ;
[ 168]       ar a1      ;
[ 169]       gr c6      ;
[ 170]       ca  0      ;
[ 171]       hv a0      ;
[ 172]       sy 64      ;
[ 173]       ar c5      ;
[ 174]       gr c6      ;
[ 175]       hv a0      ;
[ 176]c5:    qq 20.39   ;
[ 177]c6:    qq 19.39   ;


;      reset and start

[ 178]a0:    tk 42      ;
[ 179]       ar c7      ;
[ 180]       gr c8      ; reset ok-counter
[ 181]       pp  0      ;
[ 182]       ps  0      ;
[ 183]       hv i+2     ;
[ 184]a1:    qq -1.39   ; constant
[ 185]       tk 42      ;
[ 186]       hv i+2 NKB;
[ 187]       zq  LKB    ;
[ 188]       ud -2      ; test of adress, fullword
[ 189]       sr a1      ;
[ 190]       ca  0      ;
[ 191]       hv i+2     ;
[ 192]       hs 2b      ; print(ga0)
[ 193]       qqtb4XVD   ; [qq  ,hv b4], test of +half-word mark, print(H)
[ 194]      qq          ; dummy
[ 195]a4:   ar a1       ; test of LA, LB
[ 196]      hs 8b4 LA   ; print(L)
[ 197]      hs 8b4 LB   ; print(L)
[ 198]      pi -1       ; test of T, Z, K
[ 199]      hs 5b5NTC   ; print(T)
[ 200]      hs 5b6 LZ   ; print(Z)
[ 201]      tk 42       ;
[ 202]      hs 5b6 NZ   ; print(Z)
[ 203]      hs 5b4 LO   ; print(K)
[ 204]      tk 42       ; R:=0
[ 205]a5:   pm a1       ; test of X, M:=-1.39
[ 206]      cl 40  X    ;
[ 207]      hv i+2 LZ   ;
[ 208]      hs 8b5 NZ   ; print(X)
[ 209]      pm a1  X    ; R:=-1.39, M:=0
[ 210]      sr a1       ; R:=0
[ 211]      hv i+2 LZ   ;
[ 212]      hs 8b5 NZ   ; print(X)
[ 213]      tk 42  X    ; M:=0
[ 214]      xr          ; R:=0
[ 215]a6:   hv i+2 LZ   ;
[ 216]      hs 8b5 NZ   ; print(X)
[ 217]      ar a1  X    ; M:=-1.39
[ 218]      xr          ; R:=-1.39
[ 219]      sr a1       ; R:=o
[ 220]      hv i+2 LZ   ;
[ 221]      hs 8b5 NZ   ; print(X)
[ 222]      qq      V   ; test of V
[ 223]      hs 2b6      ; print(V)
[ 224]      pm a1  XV   ; test of XV
[ 225]a7:   hs b6       ; print(XV)
[ 226]      sr a1       ;
[ 227]      hv i+2 LZ   ;
[ 228]      hs b6   NZ  ; print(XV)
[ 229]      arf a1  X   ; test of floating-point mark
[ 230]      hv i+2 NZ   ;
[ 231]      hs 8b3      ; print(f)
[ 232]      tk 42       ;
[ 233]      ar a1  D    ; test of D
[ 234]      ca a1       ;
[ 235]      hv i+2      ;
[ 236]      hs 5b3      ; print(D)
[ 237]      tk 42       ;
[ 238]      ar a1       ; test of n, full-word
[ 239]a8:   qqn         ;
[ 240]      hv i+2 LZ   ;
[ 241]      hs 6b  NZ   ; print(n0)
[ 242]      ca (1b)     ; test of i, full-word
[ 243]      hv i+2      ;
[ 244]      hs 8b       ; print(i0)
[ 245]      tk 42       ;
[ 246]      ar ra1      ; test of r, full-word
[ 247]      sr a1       ;
[ 248]      hv i+2 LZ   ;
[ 249]a9:   hs 10b NZ   ; print(r0)
[ 250]      ps a1       ; test of s, full-word
[ 251]      ar s        ;
[ 252]      sr a1       ;
[ 253]      ps  0       ;
[ 254]      hv i+2 LZ   ;
[ 255]      hs 12b NZ   ; print(s0)
[ 256]      pp a1       ; test of p, full-word
[ 257]      ar  p      ;
[ 258]      sr a1      ;
[ 259]a10:  pp  0      ;
[ 260]      hv i+2 LZ  ;
[ 261]      hs 14b NZ  ; print(p0)
[ 262]      ar a1      ; test of LH hailf-word
[ 263]      ps 42      ;
[ 264]      ud    ,    ; LH, basic op
[ 265]      ps  0      ;
[ 266]      hv i+2 LZ  ;
[ 267]      hs 2b1 NZ  ; print(g1)
[ 268]      ps -1      ; test of LH adress
[ 269]a11:  ud -2,     ;
[ 270]      ps 0       ;
[ 271]      sr a1      ;
[ 272]      hv i+2 LZ  ;
[ 273]      hs 4b1 NZ  ; print(a1)
[ 274]      ar a1      ; n, LH
[ 275]      qqn  ,     ;
[ 276]      hv i+2 LZ  ;
[ 277]      hs 6b1 NZ  ; print(n1)
[ 278]      pi i-4     ;
[ 279]      gi i+1     ;
[ 280]      ar (i-6),  ; i, LH
[ 281]a12:  sr a1      ;
[ 282]      hv i+2 LZ  ;
[ 283]      hs 8b1 NZ  ; print(i1)
[ 284]      pi a1-i-2  ;
[ 285]      gi i+1     ;
[ 286]      ar ra1,    ; r, LH
[ 287]      sr a1      ;
[ 288]      hv i+2 LZ  ;
[ 289]      hs 10b1NZ  ; print(r1)
[ 290]      ps a1      ;
[ 291]      ar  s,     ;
[ 292]      sr a1      ;
[ 293]a13:  ps. 0      ;
[ 294]      hv i+2 LZ  ;
[ 295]      hs 12b1NZ  ; print(s1)
[ 296]      pp a1      ;
[ 297]      ar  p,     ;
[ 298]      sr a1      ;
[ 299]      pp  0      ;
[ 300]      hv i+2 LZ  ;
[ 301]      hs 14b1NZ  ; print(p1)
[ 302]      ,tl [VM]   ; test of -half-word
[ 303]a14:  hv i+2     ;
[ 304]      hs 1b4     ; print(h)
[ 305]      tk 42      ;
[ 306]      ar a1      ; test of RH half-word
[ 307]      ps 42      ;
[ 308]      ,ud        ; basic op
[ 309]      qq         ; dummy
[ 310]      ps  0      ;
[ 311]      hv i+2 LZ  ;
[ 312]      hs 2b2 NZ  ; print(g2)
[ 313]a15:  ps -1      ; RH adress
[ 314]      ,ud -2     ;
[ 315]      sr a1      ;
[ 316]      ps  0      ;
[ 317]      hv i+2 LZ  ;
[ 318]      hs 4b2 NZ  ; print(a2)
[ 319]      ar a1      ; n, RH
[ 320]       ,qqn       ;
[ 321]       hv i+2 LZ  ;
[ 322]       hs 6b2 NZ  ; print(n2)
[ 323]a16:   ,ca (1b)   ; i, RH
[ 324]       hv i+2     ;
[ 325]       hs 8b2     ; print(i2)
[ 326]       ,ar ra1    ; r, RH
[ 327]       sr a1      ;
[ 328]       hv i+2 LZ  ;
[ 329]       hs 10b2NZ  ; print(r2)
[ 330]       ps a1      ; s, RH
[ 331]       ,ar  s     ;
[ 332]       sr a1      ;
[ 333]a17:   ps  0      ;
[ 334]       hv i+2 LZ  ;
[ 335]       hs 12b2NZ  ; print(s2)
[ 336]       pp a1      ; p, RH
[ 337]       ,ar  p     ;
[ 338]       sr a1      ;
[ 339]       pp  0      ;
[ 340]       hv i+2 LZ  ;
[ 341]       hs 14b2NZ  ; print(p2)
[ 342]       ,tl  [VM]  ; test of -half-word mark
[ 343]a18:   hv i+2     ;
[ 344]       hs 1b4     ; print(h) .
[ 345]       pa i+1     ; test of increment
[ 346]       qq  ,qq 1  ;
[ 347]       ar i-1     ;
[ 348]       ca  0      ;
[ 349]       hv i+2     ;
[ 350]       hs b5      ; print(q)
[ 351]       tk 42      ;
[ 352]       pa i+1     ; test of -,B∧q
[ 353]a19:   ar ta1 NZ  ;
[ 354]       hv i+3 LZ  ;
[ 355]       ud b5  NZ  ; print(qB)
[ 356]       hs 2b3 NZ  ;
[ 357]       ar a1      ; test of q
[ 358]       pa i+1     ;
[ 359]       sr  ta1    ;
[ 360]       hv i+2 LZ  ;
[ 361]       hs b5  NZ  ; print(q)
[ 362]       qq tb4 XVM; [qq  ,hs b4]
[ 363]a20:   qq         ; test of +half-word mark, print(H)
[ 364]       tk 42      ;
[ 365]       ca (1b)    ; test of i, full-word
[ 366]       hv i+2     ;
[ 367]       hs 8b      ; print(i0)
[ 368]       tk 42      ; test of increment
[ 369]       ga i+12    ; decoding of TD≠0
[ 370]       ar a1      ;
[ 371]       qq(i+10)t512
[ 372]       qq(i+9)t256
[ 373]       qq(i+8)t128
[ 374]a21:   qq(i+7)t64;
[ 375]       qq(i+6)t32;
[ 376]       qq(i+5)t16;
[ 377]       qq(i+4)t8  ;
[ 378]       qq(i+3)t4  ;
[ 379]       qq(i+2)t2  ;
[ 380]       qq(i+1)t1  ;
[ 381]       ca -1      ;
[ 382]       hv i+2     ;
[ 383]      hs b5     ; print(q)
[ 384]a22:  ga i+1    ; decoding of
[ 385]      pi -1 t1  ; static op
[ 386]      tk 42     ;
[ 387]      ar i-2    ;
[ 388]      ca -1     ;
[ 389]      hv i+2    ;
[ 390]      hs 2b5    ; print(qs)
[ 391]      pi -500   ;
[ 392]      gi i+1    ; decoding of non-static op
[ 393]      bt-500t-100
[ 394]a23:  hv i+2    ;
[ 395]      hs 2b5    ; print(qs)
[ 396]      tk 42     ;
[ 397]      pa i+1    ; decoding of
[ 398]      nc 0 t-1  ; non-static op
[ 399]      hv i+2    ;
[ 400]      hs 2b5    ; print(qs)
[ 401]      qq tb4 XVM; test of +h
[ 402]      qq        ;
[ 403]      tk 42     ;
[ 404]      pa i+2    ; test of step 12
[ 405]      nt i+1    ; increment∧r
[ 406]a24:  ca  r     ;
[ 407]      hv i+3    ;
[ 408]      ud 10b    ; print(rq)
[ 409]      hs b5     ;
[ 410]      pa i+3    ;
[ 411]      ps i+2    ;
[ 412]      nt i+1    ; increment∧s
[ 413]      ca   s    ;
[ 414]      hv i+4    ;
[ 415]      ps  0     ;
[ 416]      ud 12b    ; print(sq)
[ 417]a25:  hs b5     ;
[ 418]      ps  0     ;
[ 419]      pa i+3    ;
[ 420]      pp i+2    ; increment∧p
[ 421]      nt i+1    ;
[ 422]      ca  p     ;
[ 423]      hv i+3    ;
[ 424]      ud 14b    ; print(pq)
[ 425]      hs b5     ;
[ 426]      pp  0     ;
[ 427]      tk 42     ;
[ 428]      hv 1a1 NZ ;
[ 429]a26:  ar a1     ;
[ 430]      ncn   LZ  ; test of -,B∧n
[ 431]      hv i+3    ;
[ 432]      ud 6b     ; print(nB)
[ 433]      hs 2b3    ;
[ 434]      tk 42     ;
[ 435]      ar a1     ; test of -,B∧(s)
[ 436]      ga i+1    ;
[ 437]      qq        ;
[ 438]      ps i-1    ;
[ 439]      nc (s) LZ ;
[ 440]      hv i+5    ;
[ 441]a27:  ps  0     ;
[ 442]      ud 8b     ; print(isB)
[ 443]      ud 12b    ;
[ 444]      hs 2b3    ;
[ 445]      ga i+1    ; test of (r) and (s)
[ 446]      qq         ;
[ 447]      ps i-1     ;
[ 448]      ca (s)     ;
[ 449]      hv i+4     ;
[ 450]      ps  0      ;
[ 451]a28:  ud 8b      ; print(is0)
[ 452]      hs 12b     ;
[ 453]      ps  0      ;
[ 454]      ca (r-8)   ;
[ 455]      hv i+3     ;
[ 456]      ud 8b      ; print(ir0)
[ 457]      hs 10b     ;
[ 458]a29:  pp -1      ; test of IK (mode 4)
[ 459]      pi  0      ;
[ 460]      tk 42      ;
[ 461]      nc  0 IKC  ; p:=0, in:=-1
[ 462]      hs b4      ; if +h then print(H) .
[ 463]      gp i+1     ;
[ 464]      ca         ;
[ 465]      hv i+2     ;
[ 466]a30:  hs 3b4     ; print(IK)
[ 467]      gi i+1     ;
[ 468]      qq         ;
[ 469]      nc  0 IKC  ; p:=-1, in:=0
[ 470]      hs b4      ; if +h then print(H)
[ 471]      ar i-3     ;
[ 472]      ca -1      ;
[ 473]      hv i+2     ;
[ 474]      hs 3b4     ; print(IK)
[ 475]      gp i+1     ;
[ 476]      ca         ;
[ 477]a31:  hv i+2     ;
[ 478]      hs 3b4     ; print(IK)
[ 479]      gi i+2     ;
[ 480]      tk 42      ;
[ 481]      ca         ;
[ 482]      hv i+2     ;
[ 483]      hs 3b4     ; print(IK)
[ 484]      qq     V   ; test of V
[ 485]      hs 2b6     ; print(V)
[ 486]      pp -1      ; test of VIK
[ 487]      pi  0 VIKC; p:=0, in:=-1
[ 488]      hs i+2     ; print(VIK)
[ 489]a32:  hv i+4     ;
[ 490]      ud 2b6     ;
[ 491]      ud 3b6     ;
[ 492]      hv 3b4     ;
[ 493]      tk 42      ;
[ 494]      gp i+1     ;
[ 495]      ca         ;
[ 496]      hv i+2     ;
[ 497]      hs i-7     ; print(VIK)
[ 498]      hv 1a1 NZ  ; skip if +h
[ 499]      pm a1  X   ; test of X
[ 500]      hs 8b5 LZ  ; print(X)
[ 501]      pm a1 XIKC; test of XIK
[ 502]      gp i+1     ; R:=-1, M:=0, p:=-1, in:=0
[ 503]a33:  ca         ;
[ 504]      hv i+4 NZ  ;
[ 505]      ud 8b5     ; print (XIK)
[ 506]      ud 9b5     ;
[ 507]      hs 3b4     ;
[ 508]c7:   qq 2000.39; Ok counter
[ 509]c8:  qq          ; work
[ 510]     tk 42       ;
[ 511]     ar i-2      ;
[ 5i2]     ar a1       ;
[ 513]a34: gr i-4      ;
[ 514]     hv 1a1 NZ ; if 2000 successful runs then
[ 515]     sy 62       ; print(ok)
[ 516]     sy 38       ;
[ 517]     sy 34       ;
[ 518]     sy 29       ;
[ 519]a35: hv a3       ; goto start and reset
```

e27

Rialto, 25 april 1968
JFM
t26

Test 26, Input/Output Typewriter

The program is used for test of the typewriter, when the typewriter is used as outputmedium. For test of inputfunctions one should use t9. The program is published as a binary punched papertape, which may be read to the core store by means of the primitive loaderprogram on track 0 of HJÆLP or HELP 3.

Method

The program writes lines as shown on the figure. These lines are constructed in such a way that all characters are typed and all functions (caseshift, carriage return, tabulation and ribbon shift) are activated several times. When the output on the figure has been typed the program repeats the entire process.

```
   123   456   789   0-<   qwe   rty   uio   påa
   ∨x/   =;[   ]()   ∧+>   QWE   RTY   UIO   PÅA
   123   456   789   0-<   qwe   rty   uio   påa
   ∨x/   =;[   ]()   ∧+>   QWE   RTY   UIO   PÅA
   123   456   789   0-<   qwe   rty   uio   påa
   ∨x/   =;[   ]()   ∧+>   QWE   RTY   UIO   PÅA
   123   456   789   0-<   qwe   rty   uio   påa
   ∨x/   =;[   ]()   ∧+>   QWE   RTY   UIO   PÅA
sdfg        hjkl        æøzx        cvbn        m,.-
SDFG        HJKL        ÆØZX        CVBN        M10:+
sdfg        hjkl        æøzx        cvbn        m,.-
SDFG        HJKL        ÆØZX        CVBN        M10:+
sdfg        hjkl        æøzx        cvbn        m,.-
SDFG        HJKL        ÆØZX        CVBN        M10:+
sdfg        hjkl        æøzx        cvbn        m,.-
SDFG        HJKL        ÆØZX        CVBN        M10:+
1234567890<  qwertyuiopå  asdfghjklæø  zxcvbnm,.-
∨x/=;[]()∧>  QWERTYUIOPÅ  ASDFGHJKLÆØ  ZXCVBNM10:+
1234567890<  qwertyuiopå  asdfghjklæø  zxcvbnm,.-
∨x/=;[]()∧>  QWERTYUIOPÅ  ASDFGHJKLÆØ  ZXCVBNM10:+
1234567890<  qwertyuiopå  asdfghjklæø  zxcvbnm,.-
∨x/=;[]()∧>  QWERTYUIOPÅ  ASDFGHJKLÆØ  ZXCVBNM10:+
1234567890<  qwertyuiopå  asdfghjklæø  zxcvbnm,.-
∨x/=;[]()∧>  QWERTYUIOPÅ  ASDFGHJKLÆØ  ZXCVBNM10:+
```

GIER SYSTEM LIBRARY

Order No.: 24
Class: 1.2
Type: Report
Author: Jan Flemming [illegible]
Ed.: December, 1966, [illegible]

Diagnostic Programs for the RC-3000 Converter Unit

INTRODUCTION

This paper holds a description of some catalogues and data-tapes for testing of the RC 3000 converter unit. In the first place the purpose is to shorten the time necessary for trouble-shooting. Concerning the achievement of this certain aim catalogues and data-tapes are constructed in such a way that a maximum of information is given in registers, on the output medium etc. under the circumstances already given. Some of the catalogues, and data-tapes will be especially useful for measuring on oscilloscope of dynamic phenomena; other catalogues are merely intended on the testing of some static converter functions. By planning the catalogues and data-tapes first of all a standard equipment consisting of one RC 3000 converter unit, one RC 2000 paper tape reader, and one AMPEX TM-7 tape station is considered. Optional equipment such as line-printer, card reader, paper tape puncher, plotter, and card puncher are not considered as usual equipment, but on the other hand a line-printer or an apparatus simulating line-printer will,of course, be of great help in the trouble-shooting. The description of the catalogues and data-tapes hold information about the contents of the catalogue, and about the normal (errorfree) function, and the possible reasons to errorstops or erroroutput.

Abbreviations and Terminology

The following abbreviations are so frequently used that a list of brief explanations might come in handy.

abbreviation	abbreviated from	explanation of the term
DEL	DELETE	A specialcharacter meaning that the inputcharacter in question will not produce any output.
LCS	LOWER CAT.SHIFT	A specialcharacter meaning that the addition of 256 (for reader input)or 64 (for magn.tape input) to input characters before catalogue reading will be skipped for all the input succeeding.
UCS	UPPER CAT.SHIFT	A specialcharacter meaning that addition of 256, viz. 64 shall take place on all the following input.
EB	END OF BLOCK	A specialcharacter meaning that the accumulated block has to be output.
EBS	EB WITH STOP	As EB, but with the stopping of RC 3000 after the output.

Other terms often in use are:

Catalogue which is to be used for catalogue paper tape as well as for the part of the core store which is used as catalogue.

Lower table is the part of the store which can be accessed between a LCS and a UCS.

Upper table is the part of the store which can be accessed between a UCS and a LCS.

Catalogue reading means reading of a cell in the catalogue, using an inputcharacter as address, and storing the cataloguecharacter in the blockaccumulator or skipping the cataloguecharacter (only for DEL, LCS and UCS).

Blockaccumulator is the part of the core store in which characters are collected and stored before jump to an output program.

BASA is the address of the first cell used for the blockaccumulator.

To each catalogue and data-tape are assigned a name consisting of one to three letters, a number and in some cases an additional letter group. The first letter group is built up as follows:

First letter:

C	for catalogue tape
D	for data-tape (paper tape or magnetic tape)

Second letter (if any):

R	for input from paper tape
M	for input from magnetic tape
C	for input via RC 3000 (for instance card reader)

Third letter (if any):

P	for output to printer or paper tape puncher
M	for output to magnetic tape or
C	for output to other device (card puncher)

The number following the first letter group is used as an index for tapes with equal first group. The second letter group is (when used) either BCD or CDC. BDC refers to magnetic tapes with even parity. CDC refers to a special mechanism for case shift on the line printer. One can have both BCD and CDC on the same tape.

Two special devices might be useful in trouble shooting. These are the printer simulator, and pushbutton for interruption of the parity check on transfers to and from the core store in RC 3000.

The printsimulator

Consists of a box with one switch and one pushbutton and two cables. The cables are connected to jacks J3 and J4 in the converter. The switch has the two positions RUN and STEP. In the RUN position a series of Busy and Ready signals are sent to RC 3000 which therefore can run continouesly. In the STEP position a single ready signal is sent to RC 3000 each time the pushbutton is depressed, thus allowing us to process one character and stop.

The paritybutton

The paritybutton is a pushbutton situated behind the pushbuttonpanel by the display panel (this is valid from converter No. xxx, on earlier converters the button is situated inside the frontcover at the frame. When the button is depressed the ERROR flip-flop can not be set for parity errors in the IR-register, but ERROR can still be set for length-errors in the catalogue tape etc. Using this button it is possible to read in catalogues containing characters with even parity. This feature is very useful if one want to stop RC 3000 at the very moment an error occures in a dynamic run.

CRM - 0, CRM - 1, CRM - 2, CRM - 3, CRM - 4, CRM - 5, CRM - 6, CRM - 7

These eight catalogues all have the length 512. Input is taken from paper tape and output is sent to magnetic tape and written with odd parity. The catalogues have the same character in all cells from 1 - 255, the cells 0 and 256 - 511 contains EBS.

Catalogue	Catalogue contents (1 - 255)
CRM - 1	1
CRM - 2	2
CRM - 3	4
CRM - 4	8
CRM - 5	16
CRM - 6	32
CRM - 7	0
CRM - 0	64

The data used is D - 1. In this way we obtain a constant information on the outputlines. The reason for the EBS's in all the not accessible cells up to 511 is that in this way an erroneous bit in the AR-register position 8 or zeroes in all positions of AR will cause a stop of RC-3000. The catalogue CRM - 7 will only write the parity bit on the magnetic tape, and CRM - 0 will give parity error on the magnetic tape. (The PET lamp will be lit). This means that with CRM - 7 one can test whether the parity bit is generated or not when it is present in the core store. With CRM - 0 one can test whether a parity bit is generated or not when it is not present in the core store.

CMP - 0

Catalogue for test of the magnetic tape produced by CRM - 1, CRM - 2, CRM - 3, CRM - 4, CRM - 5, CRM - 6 or CRM - 7. The output is sent to the line printer or the line printer simulator. The output is the number of the track on the magnetic tape (i.e. it is 1, 2, 3, 4, 5, 6 or 7). The catalogue is read in using the parity button and place parity errors in all cells except 0, 1, 2, 3, 4, 5 and 6. In this way a character from the magnetic tape will cause an immediate stop in the ERROR situation and in the AR-register the character which caused the stop can be seen.

The IR-register will contain in case of error the number of cell which caused the error using position 7 in the cell as parity bit. Therefore, normally, the contents of IR and AR ahould be equal. If not there must be some error in connection with the AR-register (e.g. the decoding), or an error in one or more positions or IR.

CMP - 1

As CMP - 0 but with EBS in all irrelevant cells. With this catalogue, which can be read in on all converters, one will have a stop after an error. In the stop situation the information displayed in the registers have no connection with the error, one can only register that there has been some sort of error. This is one of the reasons for the special parity button.

C - 0

Catalogue for test of the core store. The catalogue contents is 8 sets of characters each set having characters running from 0 to 127 with odd parity. To this 8 sets is added a character with value 127, so that we have in all 128 x 8 + 1 = 1025. The catalogue is constructed as an infinite loop. In this way we are able to displace the first character one cell in each loop and after 127 loops all legal bit combinations have been written in all cells, and the BASA has been set to all values from 0 to 1023. When the running is stopped by activating UP on the reader, the normal reaction will be that ERROR is lit because in only approx. 3 % of the cases we will have read a multiple of 32 characters. If ERROR is lit during the run, and the converter stops in program 1, step 5 with even parity in the IR-register, the error was a parity error from the paper tape (i.e. an error on the paper tape or an error in the transfer from RC 2000 to the converter). For converters with serial number from 3122 stop with program 1, step 6 selected may occur. In this case we have an error in the core store, and the cell number can be seen in the AR-register, and the wrong cell contents in the IR-register. If the button "proceed read catalogue" is depressed the read of the catalogue tape will be continued.

C - 1

Catalogue for testing of counting in the OC-register, which is the only counting register in RC 3000 and vital for the correct function. The catalogue is a tape with 1 + 1023 characters. The first character is punched about one inch in front of the second and following characters. The characters used all have the one hole in track 5. If one read in the entire catalogue the following indicators should be lit: PAPERTAPE INPUT, MAGN. TAPE OUTPUT BINARY. All positions in OC except OC 10 should be zero. Of one reads in the catalogue starting with the second character ERROR should be lit (on converters from No. 3122 LENGTH ERROR), and the OR-register shall have ones in position 0 - 9 and zero in position 10.

CRM - 8

Catalogue for testing of the special character EBS. Each second cell of the catalogue contains EBS (i.e. EBS in cell 1, 3, 5, ...251). The other cells contains the values 1 - 63 in the following way:

Cell	Value
0	1
2	2
4	3
-	-
-	-
124	63
126	1
128	2
-	-
250	63

The catalogue is used in connection with the data tape D - 10 which will refer to cell 1 - 2 - 3 up to 251 and then again to 1, 2, 3 etc. This means that each second character will give EBS and the output block (consisting of 512 characters) will be:

data	block
1	1-1-1-....1
3	2-1-1-....1
5	3-1-1-....1
7	4-1-1-....1
-	-
125	63-1-1-....1
127	1-1-1-....1
129	2-1-1-....1
-	-
250	63-1-1-....1

The output to the magnetic tape can be checked to some degree with the catalogue CMP - 2, but the best check can be performed by a computer program. This program shall read the blocks one by one and check that the first character is the block number modulo 64, and that the rest of the block is ones. If the rest of the block is different from ones the reason could be that EBS does not act correctly. If all blocks consist of ones the EBS is derived also when the catalogue cell contains one of the values 1 - 63.

CRM - 8 BCD

Catalogue with functions as CRM - 8, but writing the magnetic tape with even parity.

CMP - 2

Catalogue for printing of tapes produced by CRM - 8. The catalogues output is one character per block and this character will be the value of the first character in the block, but ones will be neglected at output.

CMP - 2 BCD

Catalogue with functions as CMP - 2, but printing of tapes produced by CRM - 8 BCD.

CRM - 9

Catalogue for testing of the special characters DELETE, LOWER CATALOGUE SHIFT and UPPER CATALOGUE SHIFT. The lower table part of the catalogue is stored in cells 1 - 255 and the upper table part in cells 257 - 511. BASA is set to 1023. Therefore a single character which is not decoded as special character will produce output. Below is sketched how the catalogue is built up:

Value of input character	Lower Table cell	Upper Table cell
not existing	0 parity	0 + 256 parity
1	1 UCS	1 + 256 parity
2	2 parity	2 + 256 DEL
3	3 parity	3 + 256 LCS
4	4 DEL	4 + 256 parity
5	5 UCS	5 + 256 parity
6	6 parity	6 + 256 DEL
7	7 parity	7 + 256 LCS
8	8 DEL	8 + 256 parity
-		
-		
254	254 parity	254 + 256 DEL
255	255 parity	255 + 256 LCS
512	all contains parity	

Parity means that the cell has even parity, so that an erroneous reference to the cell will stop the converter in the ERROR-situation. As seen fron the table each input character will refer to a cell with some special character and if LCS and UCS do not act correct, reference to a cell with parity error will be made, and the converter stops in the ERROR-situation with the input character in the AR-register positions 0 - 7 and AR 8 = 1, if 256 has been added in the lower table case and AR 8 = 0, if 256 has not been added in the upper table case.

CRM - 10

As CRM - 9, but EBS instead of parity errors in the erroneous cells.

CMP - 3

Catalogue for test of LCS and UCS when input is taken from magnetic tape. The lower table part is stored in cells 1 - 63 and the upper table part is stored from 65 - 127. In this case addition of 64 will take place when the UCS is derived. The catalogue is built up as follows:

Value of input	Lower Table		Upper Table	
	cell No.	contents	cell No.	contents
not existing	0	parity	0 + 64	parity
1	1	UCS	1 + 64	parity
2	2	parity	2 + 64	DEL
3	3	parity	3 + 64	LCS
4	4	DEL	4 + 64	parity
5	5	UCS	5 + 64	parity
6	6	parity	6 + 64	DEL
7	7	parity	7 + 64	LCS
8	8	DEL	8 + 64	parity
-	-	-	-	-
-	-	-	-	-
62	62	parity	62 + 64	DEL
62	63	parity	63 + 64	LCS
not existing	all contains parity			

In case of error (i.e. ERROR it lit) the AR-register positions 0 - 5 contains the inputcharacter and AR 6 = 1, if 64 has been added in the lower table case and AR 6 = 0, if 64 has not been added in the upper table case.

CMP - 7

As CMP - 3, but with EBS instead of parity errors.

CMP - 17

Catalogue for production of magnetic tape to be used in connection with CMP - 7. The data tape used is D - 1. The blocks produced consist of the 63 values 1 - 63. The catalogue is built up as follows:

value fo input	output
1 - 63	1 - 63
64 - 125	1 - 63
127 - 189	1 - 63
190 - 252	1 - 63
253	DEL
254	DEL
255	DEL

BASA is to obtain the correct block length, set to 961.

A few catalogues have been constructed for testing the selection of input and output and conversion mode. These catalogues all have the length 32. The necessary catalogues are listed below:

CATALOGUE	INPUT LAMPS	OUTPUT LAMPS	MODE	CU
C2	Magnetic tape	Printer	CDC, BCD	111111
C3	Paper tape	Magnetic tape	Binary	000000

As seen from the list the catalogues are clear nonsense from the users point of view, they can only be used as test of the controlunit.

CRP - 1

Catalogue for test of the line printed. The data tape D - 2 is used, and the correct printout is given in the appendix.

CRP - 1 CDC

Catalogue for test of printout in the special CDC - mode. The data tape used is D - 3. The correct printout is given in the appendix.

CRM - 11, CRM - 12, CRM - 13, CRM - 15, CRM - 16

Catalogues for test of reading of OLIVETTI six track paper tape. Each catalogue test reading of one track and the output to magnetic tape will be the value of that track in the following manner:

catalogue	test of track	output	from catalogue cell No.	data tape
CRM - 11	A	1	1	D4
CRM - 12	B	2	2	D5
CRM - 13	C	4	4	D6
CRM - 14	D	8	8	D7
CRM - 15	E	16	16	D8
CRM - 16	F	32	32	D9

All other cells in the catalogue than the test-cell with the output information contain parity errors and references to these due to erroneous reading will cause the converter to stop in ERROR. In this moment contents of the AR-register is the erroneous character.

C - 4

Catalogue tape for test of OLIVETTI reading. The tape should be read with the parity button depressed. After reading of the tape ERROR shall remain unlit, otherwise there was a length error, and the number of missing characters can be calculated as

missing = 32 - (contents of OC 0 - 4)

This number is the number of missing characters modulo 32.

D - 1

A 8-track data tape consisting of four sets of characters. Each set holds the characters with values from 1 up to 255. The paper tape is an infinite loop. Using this tape as data for the catalogues, one obtain that references to all cells in the catalogue can be made. Cell 0 will be the only cell to which no reference can be made.

D - 2

Data tape for test of the line printer. The tape is used in connection with some of the catalogues and will give output of all types of relevant information to all print positions on the line printer.

D - 3

Data tape for test of the CDC printing on the line printer.

D - 4, D - 5, D - 6, D - 7, D - 8, D - 9

Infinite data tapes for test of reading of OLIVETTI six track paper tape. Each tape has holes on one single track.

D - 10

Data tape for CRM - 8 and CRM - 8 BCD. This infinite tape consists of four sets of characters each set running from 1 to 251.

B U F F E R T E S T 2   (A 203)

```
hp-knap  121
1              [illegible]                   KA = 1
klar                                         repeat with same parameters
1w3
ae4 } parametre
aed


0000                                         number of errors by = 16 (eller By 8)

0000
0000

0000000000x0000000000x0000000101x0111001000x00
0000000001x0000000000x0000000101x0111001000x00  bf adr 0001

0000000000x0000000000x0000010000x0100101001x00
0000000001x0000000000x0000010000x0100101001x00  bf adr 0002
```

AD 9·10·11

AD 6·7·8

AD 3·4·5

AD 0·1·2

1. correct contens of the celle
2.read out contens

bf adr octal code

by = 8

start r1 := 512

Stop r1 := 525 hvis $\overline{KB}$

```
0
hp-knap  -499                  C D C 1      (TAPETEST 1)
1    [illegible]
st. nr=1
klar
wer }
dfr } parametre
drf }

status  000011 001111 111111 110000 000000 000000 000000

p.  0000
0000
st. nr=2
klar
as }
sd } parametre
sd }

status  000011 001111 111111 110000 000000 000000 000000

p.  0000
0000
status  100011 000010 110101 000000 000000 000000 000000

p.  0001

000000 100011 111101 011110 000010 110111 001000
000000 100010 111101 011110 000010 110111 001000  bf adr 1323

000011 000000 000001 100000 101001 010010 110000
000000 000000 000000 000000 000011 000000 000000  bf adr 1324

000010 111110 001110 011110 000001 001001 110000
000000 000000 000000 000000 000000 000000 000000  bf adr 1325
```

as described at buffertest 2

p. = number of par. errors

st. nr. = 1 - 2 - 3   start r1 := 512

BUFFERTEST - 3

Program for test of the buffer store according to the principles of test B10 and test B11.

INPUT AND START

~~The program is published in condensed form to be read in by track 0.~~
After input the program types ⟨buffertest - 3⟩ and stops if KB is one. In the stopsituation it is possible to sct the bits 0 - 39 of the R-reg. and RA and RB in the Indicator then these bits will be used in the test.

BUILD IN TEST BITS

The build in test bits is taken in sequence from an array of the following form

$$\underbrace{1,1,1,1,\ldots.1}_{42\text{ bits}},\underbrace{0,0,0,0,\ldots.0}_{42\text{ bits}},\underbrace{1,1,1,1,\ldots.1}_{41\text{ bits}}$$

This means that each cell is tested with 84 different bit patterns.

ERROR REACTIONS

The ncrmal condition for KA is KA=0. In this case there will only be reaction from the program at the end of one complete testcycle.

The reaction will be:

NO ERRORS: The program types ⟨CR, ok⟩ and hoots in 5 second with the frequency 1000 c/s.

ERROR: The program types ⟨CR, -⟩ and hoots in 5 second with alternating frequency (950 c/s and 1100 c/s).

KA=1 will cause the program to stop when an error is detected. There is four possible stop points in the program:

1) rl = 138
   Informationerror, R: = bufferaddress 39

2) rl = 140
   Informationerror, R: = wrong information
   M: = correct information

3) rl = 143
   Error in marks, R 8-9: = correct marks,
   In RA, RB : = wrong marks

4) rl = 148
   Error in marks, R: = bufferaddress 39.

Jan F. Madsen
16.I.67

INTEGER <-> BINARY
CONVERSION

0	....	-1024	50	....11..1.	-974	100	...11..1..	-924
1	1	-1023	51	....11..11	-973	101	...11..1.1	-923
2	........1.	-1022	52	....11.1..	-972	102	...11..11.	-922
3	11	-1021	53	....11.1.1	-971	103	...11..111	-921
4	.......1..	-1020	54	....11.11.	-970	104	...11.1...	-920
5	.......1.1	-1019	55	....11.111	-969	105	...11.1..1	-919
6	.......11.	-1018	56	....111...	-968	106	...11.1.1.	-918
7	111	-1017	57	....111..1	-967	107	...11.1.11	-917
8	......1...	-1016	58	....111.1.	-966	108	...11.11..	-916
9	......1..1	-1015	59	....111.11	-965	109	...11.11.1	-915
10	......1.1.	-1014	60	....1111..	-964	110	...11.111.	-914
11	......1.11	-1013	61	....1111.1	-963	111	...11.1111	-913
12	......11..	-1012	62	....11111.	-962	112	...111....	-912
13	......11.1	-1011	63	111111	-961	113	...111...1	-911
14	......111.	-1010	64	...1......	-960	114	...111..1.	-910
15	1111	-1009	65	...1.....1	-959	115	...111..11	-909
16	.....1....	-1008	66	...1....1.	-958	116	...111.1..	-908
17	.....1...1	-1007	67	...1....11	-957	117	...111.1.1	-907
18	.....1..1.	-1006	68	...1...1..	-956	118	...111.11.	-906
19	.....1..11	-1005	69	...1...1.1	-955	119	...111.111	-905
20	.....1.1..	-1004	70	...1...11.	-954	120	...1111...	-904
21	.....1.1.1	-1003	71	...1...111	-953	121	...1111..1	-903
22	.....1.11.	-1002	72	...1..1...	-952	122	...1111.1.	-902
23	.....1.111	-1001	73	...1..1..1	-951	123	...1111.11	-901
24	.....11...	-1000	74	...1..1.1.	-950	124	...11111..	-900
25	.....11..1	-999	75	...1..1.11	-949	125	...11111.1	-899
26	.....11.1.	-998	76	...1..11..	-948	126	...111111.	-898
27	.....11.11	-997	77	...1..11.1	-947	127	...1111111	-897
28	.....111..	-996	78	...1..111.	-946	128	..1.......	-896
29	.....111.1	-995	79	...1..1111	-945	129	..1......1	-895
30	.....1111.	-994	80	...1.1....	-944	130	..1.....1.	-894
31	11111	-993	81	...1.1...1	-943	131	..1.....11	-893
32	....1.....	-992	82	...1.1..1.	-942	132	..1....1..	-892
33	....1....1	-991	83	...1.1..11	-941	133	..1....1.1	-891
34	....1...1.	-990	84	...1.1.1..	-940	134	..1....11.	-890
35	....1...11	-989	85	...1.1.1.1	-939	135	..1....111	-889
36	....1..1..	-988	86	...1.1.11.	-938	136	..1...1...	-888
37	....1..1.1	-987	87	...1.1.111	-937	137	..1...1..1	-887
38	....1..11.	-986	88	...1.11...	-936	138	..1...1.1.	-886
39	....1..111	-985	89	...1.11..1	-935	139	..1...1.11	-885
40	....1.1...	-984	90	...1.11.1.	-934	140	..1...11..	-884
41	....1.1..1	-983	91	...1.11.11	-933	141	..1...11.1	-883
42	....1.1.1.	-982	92	...1.111..	-932	142	..1...111.	-882
43	....1.1.11	-981	93	...1.111.1	-931	143	..1...1111	-881
44	....1.11..	-980	94	...1.1111.	-930	144	..1..1....	-880
45	....1.11.1	-979	95	...1.11111	-929	145	..1..1...1	-879
46	....1.111.	-978	96	...11.....	-928	146	..1..1..1.	-878
47	....1.1111	-977	97	...11....1	-927	147	..1..1..11	-877
48	....11....	-976	98	...11...1.	-926	148	..1..1.1..	-876
49	....11...1	-975	99	...11...11	-925	149	..1..1.1.1	-875

```
150 ..1..1.11.  -874         200 ..11..1...  -824         250 ..11111.1.  -774
151 ..1..1.111  -873         201 ..11..1..1  -823         251 ..11111.11  -773
152 ..1..11...  -872         202 ..11..1.1.  -822         252 ..111111..  -772
153 ..1..11..1  -871         203 ..11..1.11  -821         253 ..111111.1  -771
154 ..1..11.1.  -870         204 ..11..11..  -820         254 ..1111111.  -770
155 ..1..11.11  -869         205 ..11..11.1  -819         255 ..11111111  -769
156 ..1..111..  -868         206 ..11..111.  -818         256 .1........  -768
157 ..1..111.1  -867         207 ..11..1111  -817         257 .1.......1  -767
158 ..1..1111.  -866         208 ..11.1....  -816         258 .1......1.  -766
159 ..1..11111  -865         209 ..11.1...1  -815         259 .1......11  -765
160 ..1.1.....  -864         210 ..11.1..1.  -814         260 .1.....1..  -764
161 ..1.1....1  -863         211 ..11.1..11  -813         261 .1.....1.1  -763
162 ..1.1...1.  -862         212 ..11.1.1..  -812         262 .1.....11.  -762
163 ..1.1...11  -861         213 ..11.1.1.1  -811         263 .1.....111  -761
164 ..1.1..1..  -860         214 ..11.1.11.  -810         264 .1....1...  -760
165 ..1.1..1.1  -859         215 ..11.1.111  -809         265 .1....1..1  -759
166 ..1.1..11.  -858         216 ..11.11...  -808         266 .1....1.1.  -758
167 ..1.1..111  -857         217 ..11.11..1  -807         267 .1....1.11  -757
168 ..1.1.1...  -856         218 ..11.11.1.  -806         268 .1....11..  -756
169 ..1.1.1..1  -855         219 ..11.11.11  -805         269 .1....11.1  -755
170 ..1.1.1.1.  -854         220 ..11.111..  -804         270 .1....111.  -754
171 ..1.1.1.11  -853         221 ..11.111.1  -803         271 .1....1111  -753
172 ..1.1.11..  -852         222 ..11.1111.  -802         272 .1...1....  -752
173 ..1.1.11.1  -851         223 ..11.11111  -801         273 .1...1...1  -751
174 ..1.1.111.  -850         224 ..111.....  -800         274 .1...1..1.  -750
175 ..1.1.1111  -849         225 ..111....1  -799         275 .1...1..11  -749
176 ..1.11....  -848         226 ..111...1.  -798         276 .1...1.1..  -748
177 ..1.11...1  -847         227 ..111...11  -797         277 .1...1.1.1  -747
178 ..1.11..1.  -846         228 ..111..1..  -796         278 .1...1.11.  -746
179 ..1.11..11  -845         229 ..111..1.1  -795         279 .1...1.111  -745
180 ..1.11.1..  -844         230 ..111..11.  -794         280 .1...11...  -744
181 ..1.11.1.1  -843         231 ..111..111  -793         281 .1...11..1  -743
182 ..1.11.11.  -842         232 ..111.1...  -792         282 .1...11.1.  -742
183 ..1.11.111  -841         233 ..111.1..1  -791         283 .1...11.11  -741
184 ..1.111...  -840         234 ..111.1.1.  -790         284 .1...111..  -740
185 ..1.111..1  -839         235 ..111.1.11  -789         285 .1...111.1  -739
186 ..1.111.1.  -838         236 ..111.11..  -788         286 .1...1111.  -738
187 ..1.111.11  -837         237 ..111.11.1  -787         287 .1...11111  -737
188 ..1.1111..  -836         238 ..111.111.  -786         288 .1..1.....  -736
189 ..1.1111.1  -835         239 ..111.1111  -785         289 .1..1....1  -735
190 ..1.11111.  -834         240 ..1111....  -784         290 .1..1...1.  -734
191 ..1.111111  -833         241 ..1111...1  -783         291 .1..1...11  -733
192 ..11......  -832         242 ..1111..1.  -782         292 .1..1..1..  -732
193 ..11.....1  -831         243 ..1111..11  -781         293 .1..1..1.1  -731
194 ..11....1.  -830         244 ..1111.1..  -780         294 .1..1..11.  -730
195 ..11....11  -829         245 ..1111.1.1  -779         295 .1..1..111  -729
196 ..11...1..  -828         246 ..1111.11.  -778         296 .1..1.1...  -728
197 ..11...1.1  -827         247 ..1111.111  -777         297 .1..1.1..1  -727
198 ..11...11.  -826         248 ..11111...  -776         298 .1..1.1.1.  -726
199 ..11...111  -825         249 ..11111..1  -775         299 .1..1.1.11  -725
```

```
300 .1..1.11..  -724     350 .1.1.1111.  -674     400 .11..1....  -624
301 .1..1.11.1  -723     351 .1.1.11111  -673     401 .11..1...1  -623
302 .1..1.111.  -722     352 .1.11.....  -672     402 .11..1..1.  -622
303 .1..1.1111  -721     353 .1.11....1  -671     403 .11..1..11  -621
304 .1..11....  -720     354 .1.11...1.  -670     404 .11..1.1..  -620
305 .1..11...1  -719     355 .1.11...11  -669     405 .11..1.1.1  -619
306 .1..11..1.  -718     356 .1.11..1..  -668     406 .11..1.11.  -618
307 .1..11..11  -717     357 .1.11..1.1  -667     407 .11..1.111  -617
308 .1..11.1..  -716     358 .1.11..11.  -666     408 .11..11...  -616
309 .1..11.1.1  -715     359 .1.11..111  -665     409 .11..11..1  -615
310 .1..11.11.  -714     360 .1.11.1...  -664     410 .11..11.1.  -614
311 .1..11.111  -713     361 .1.11.1..1  -663     411 .11..11.11  -613
312 .1..111...  -712     362 .1.11.1.1.  -662     412 .11..111..  -612
313 .1..111..1  -711     363 .1.11.1.11  -661     413 .11..111.1  -611
314 .1..111.1.  -710     364 .1.11.11..  -660     414 .11..1111.  -610
315 .1..111.11  -709     365 .1.11.11.1  -659     415 .11..11111  -609
316 .1..1111..  -708     366 .1.11.111.  -658     416 .11.1.....  -608
317 .1..1111.1  -707     367 .1.11.1111  -657     417 .11.1....1  -607
318 .1..11111.  -706     368 .1.111....  -656     418 .11.1...1.  -606
319 .1..111111  -705     369 .1.111...1  -655     419 .11.1...11  -605
320 .1.1......  -704     370 .1.111..1.  -654     420 .11.1..1..  -604
321 .1.1.....1  -703     371 .1.111..11  -653     421 .11.1..1.1  -603
322 .1.1....1.  -702     372 .1.111.1..  -652     422 .11.1..11.  -602
323 .1.1....11  -701     373 .1.111.1.1  -651     423 .11.1..111  -601
324 .1.1...1..  -700     374 .1.111.11.  -650     424 .11.1.1...  -600
325 .1.1...1.1  -699     375 .1.111.111  -649     425 .11.1.1..1  -599
326 .1.1...11.  -698     376 .1.1111...  -648     426 .11.1.1.1.  -598
327 .1.1...111  -697     377 .1.1111..1  -647     427 .11.1.1.11  -597
328 .1.1..1...  -696     378 .1.1111.1.  -646     428 .11.1.11..  -596
329 .1.1..1..1  -695     379 .1.1111.11  -645     429 .11.1.11.1  -595
330 .1.1..1.1.  -694     380 .1.11111..  -644     430 .11.1.111.  -594
331 .1.1..1.11  -693     381 .1.11111.1  -643     431 .11.1.1111  -593
332 .1.1..11..  -692     382 .1.111111.  -642     432 .11.11....  -592
333 .1.1..11.1  -691     383 .1.1111111  -641     433 .11.11...1  -591
334 .1.1..111.  -690     384 .11.......  -640     434 .11.11..1.  -590
335 .1.1..1111  -689     385 .11......1  -639     435 .11.11..11  -589
336 .1.1.1....  -688     386 .11.....1.  -638     436 .11.11.1..  -588
337 .1.1.1...1  -687     387 .11.....11  -637     437 .11.11.1.1  -587
338 .1.1.1..1.  -686     388 .11....1..  -636     438 .11.11.11.  -586
339 .1.1.1..11  -685     389 .11....1.1  -635     439 .11.11.111  -585
340 .1.1.1.1..  -684     390 .11....11.  -634     440 .11.111...  -584
341 .1.1.1.1.1  -683     391 .11....111  -633     441 .11.111..1  -583
342 .1.1.1.11.  -682     392 .11...1...  -632     442 .11.111.1.  -582
343 .1.1.1.111  -681     393 .11...1..1  -631     443 .11.111.11  -581
344 .1.1.11...  -680     394 .11...1.1.  -630     444 .11.1111..  -580
345 .1.1.11..1  -679     395 .11...1.11  -629     445 .11.1111.1  -579
346 .1.1.11.1.  -678     396 .11...11..  -628     446 .11.11111.  -578
347 .1.1.11.11  -677     397 .11...11.1  -627     447 .11.111111  -577
348 .1.1.111..  -676     398 .11...111.  -626     448 .111......  -576
349 .1.1.111.1  -675     399 .11...1111  -625     449 .111.....1  -575
```

```
450 .111....1.   -574          500 .11111.1..   -524          550 1...1..11.   -474
451 .111....11   -573          501 .11111.1.1   -523          551 1...1..111   -473
452 .111...1..   -572          502 .11111.11.   -522          552 1...1.1...   -472
453 .111...1.1   -571          503 .11111.111   -521          553 1...1.1..1   -471
454 .111...11.   -570          504 .111111...   -520          554 1...1.1.1.   -470
455 .111...111   -569          505 .111111..1   -519          555 1...1.1.11   -469
456 .111..1...   -568          506 .111111.1.   -518          556 1...1.11..   -468
457 .111..1..1   -567          507 .111111.11   -517          557 1...1.11.1   -467
458 .111..1.1.   -566          508 .1111111..   -516          558 1...1.111.   -466
459 .111..1.11   -565          509 .1111111.1   -515          559 1...1.1111   -465
460 .111..11..   -564          510 .11111111.   -514          560 1...11....   -464
461 .111..11.1   -563          511 .111111111   -513          561 1...11...1   -463
462 .111..111.   -562          512 1.........   -512          562 1...11..1.   -462
463 .111..1111   -561          513 1........1   -511          563 1...11..11   -461
  4 .111.1....   -560          514 1.......1.   -510          564 1...11.1..   -460
465 .111.1...1   -559          515 1.......11   -509          565 1...11.1.1   -459
466 .111.1..1.   -558          516 1......1..   -508          566 1...11.11.   -458
467 .111.1..11   -557          517 1......1.1   -507          567 1...11.111   -457
468 .111.1.1..   -556          518 1......11.   -506          568 1...111...   -456
469 .111.1.1.1   -555          519 1......111   -505          569 1...111..1   -455
470 .111.1.11.   -554          520 1.....1...   -504          570 1...111.1.   -454
471 .111.1.111   -553          521 1.....1..1   -503          571 1...111.11   -453
472 .111.11...   -552          522 1.....1.1.   -502          572 1...1111..   -452
473 .111.11..1   -551          523 1.....1.11   -501          573 1...1111.1   -451
474 .111.11.1.   -550          524 1.....11..   -500          574 1...11111.   -450
475 .111.11.11   -549          525 1.....11.1   -499          575 1...111111   -449
476 .111.111..   -548          526 1.....111.   -498          576 1..1......   -448
477 .111.111.1   -547          527 1.....1111   -497          577 1..1.....1   -447
478 .111.1111.   -546          528 1....1....   -496          578 1..1....1.   -446
479 .111.11111   -545          529 1....1...1   -495          579 1..1....11   -445
480 .1111.....   -544          530 1....1..1.   -494          580 1..1...1..   -444
481 .1111....1   -543          531 1....1..11   -493          581 1..1...1.1   -443
482 .1111...1.   -542          532 1....1.1..   -492          582 1..1...11.   -442
  3 .1111...11   -541          533 1....1.1.1   -491          583 1..1...111   -441
484 .1111..1..   -540          534 1....1.11.   -490          584 1..1..1...   -440
485 .1111..1.1   -539          535 1....1.111   -489          585 1..1..1..1   -439
486 .1111..11.   -538          536 1....11...   -488          586 1..1..1.1.   -438
487 .1111..111   -537          537 1....11..1   -487          587 1..1..1.11   -437
488 .1111.1...   -536          538 1....11.1.   -486          588 1..1..11..   -436
489 .1111.1..1   -535          539 1....11.11   -485          589 1..1..11.1   -435
490 .1111.1.1.   -534          540 1....111..   -484          590 1..1..111.   -434
491 .1111.1.11   -533          541 1....111.1   -483          591 1..1..1111   -433
492 .1111.11..   -532          542 1....1111.   -482          592 1..1.1....   -432
493 .1111.11.1   -531          543 1....11111   -481          593 1..1.1...1   -431
494 .1111.111.   -530          544 1...1.....   -480          594 1..1.1..1.   -430
495 .1111.1111   -529          545 1...1....1   -479          595 1..1.1..11   -429
496 .11111....   -528          546 1...1...1.   -478          596 1..1.1.1..   -428
497 .11111...1   -527          547 1...1...11   -477          597 1..1.1.1.1   -427
498 .11111..1.   -526          548 1...1..1..   -476          598 1..1.1.11.   -426
499 .11111..11   -525          549 1...1..1.1   -475          599 1..1.1.111   -425
```

```
600 1..1.11...   -424
601 1..1.11..1   -423
602 1..1.11.1.   -422
603 1..1.11.11   -421
604 1..1.111..   -420
605 1..1.111.1   -419
606 1..1.1111.   -418
607 1..1.11111   -417
608 1..11.....   -416
609 1..11,...1   -415
610 1..11...1.   -414
611 1..11...11   -413
612 1..11..1..   -412
613 1..11..1.1   -411
614 1..11..11.   -410
615 1..11..111   -409
616 1..11.1...   -408
617 1..11.1..1   -407
618 1..11.1.1.   -406
619 1..11.1.11   -405
620 1..11.11..   -404
621 1..11.11.1   -403
622 1..11.111.   -402
623 1..11.1111   -401
624 1..111....   -400
625 1..111...1   -399
626 1..111..1.   -398
627 1..111..11   -397
628 1..111.1..   -396
629 1..111.1.1   -395
630 1..111.11.   -394
631 1..111.111   -393
632 1..1111...   -392
633 1..1111..1   -391
634 1..1111.1.   -390
635 1..1111.11   -389
636 1..11111..   -388
637 1..11111.1   -387
638 1..111111.   -386
639 1..1111111   -385
640 1.1.......   -384
641 1.1......1   -383
642 1.1.....1.   -382
643 1.1.....11   -381
644 1.1....1..   -380
645 1.1....1.1   -379
646 1.1....11.   -378
647 1.1....111   -377
648 1.1...1...   -376
649 1.1...1..1   -375
```

```
650 1.1...1.1.   -374
651 1.1...1.11   -373
652 1.1...11..   -372
653 1.1...11.1   -371
654 1.1..111.    -370
655 1.1...1111   -369
656 1.1..1....   -368
657 1.1..1...1   -367
658 1.1..1..1.   -366
659 1.1..1..11   -365
660 1.1..1.1..   -364
661 1.1..1.1.1   -363
662 1.1..1.11.   -362
663 1.1..1.111   -361
664 1.1..11...   -360
665 1.1..11..1   -359
666 1.1..11.1.   -358
667 1.1..11.11   -357
668 1.1..111..   -356
669 1.1..111.1   -355
670 1.1..1111.   -354
671 1.1..11111   -353
672 1.1.1.....   -352
673 1.1.1....1   -351
674 1.1.1...1.   -350
675 1.1.1...11   -349
676 1.1.1..1..   -348
677 1.1.1..1.1   -347
678 1.1.1..11.   -346
679 1.1.1..111   -345
680 1.1.1.1...   -344
681 1.1.1.1..1   -343
682 1.1.1.1.1.   -342
683 1.1.1.1.11   -341
684 1.1.1.11..   -340
685 1.1.1.11.1   -339
686 1.1.1.111.   -338
687 1.1.1.1111   -337
688 1.1.11....   -336
689 1.1.11...1   -335
690 1.1.11..1.   -334
691 1.1.11..11   -333
692 1.1.11.1..   -332
693 1.1.11.1.1   -331
694 1.1.11.11.   -330
695 1.1.11.111   -329
696 1.1.111...   -328
697 1.1.111..1   -327
698 1.1.111.1.   -326
699 1.1.111.11   -325
```

```
700 1.1.1111..   -324
701 1.1.1111.1   -323
702 1.1.11111.   -322
703 1.1.111111   -321
704 1.11......   -320
705 1.11.....1   -319
706 1.11....1.   -318
707 1.11....11   -317
708 1.11...1..   -316
709 1.11...1.1   -315
710 1.11...11.   -314
711 1.11...111   -313
712 1.11..1...   -312
713 1.11..1..1   -311
714 1.11..1.1.   -310
715 1.11..1.11   -309
716 1.11..11..   -308
717 1.11..11.1   -307
718 1.11..111.   -306
719 1.11..1111   -305
720 1.11.1....   -304
721 1.11.1...1   -303
722 1.11.1..1.   -302
723 1.11.1..11   -301
724 1.11.1.1..   -300
725 1.11.1.1.1   -299
726 1.11.1.11.   -298
727 1.11.1.111   -297
728 1.11.11...   -296
729 1.11.11..1   -295
730 1.11.11.1.   -294
731 1.11.11.11   -293
732 1.11.111..   -292
733 1.11.111.1   -291
734 1.11.1111.   -290
735 1.11.11111   -289
736 1.111.....   -288
737 1.111....1   -287
738 1.111...1.   -286
739 1.111...11   -285
740 1.111..1..   -284
741 1.111..1.1   -283
742 1.111..11.   -282
743 1.111..111   -281
744 1.111.1...   -280
745 1.111.1..1   -279
746 1.111.1.1.   -278
747 1.111.1.11   -277
748 1.111.11..   -276
749 1.111.11.1   -275
```

```
750 1.111.111.   -274        800 11..1.....   -224        850 11,1.1,,1,   -174
751 1.111.1111   -273        801 11..1....1   -223        851 11,1.1,,11   -173
752 1.1111....   -272        802 11..1...1.   -222        852 11,1.1,1,,   -172
753 1.1111...1   -271        803 11..1...11   -221        853 11,1.1,1,1   -171
754 1.1111..1.   -270        804 11..1..1..   -220        854 11,1.1,11,   -170
755 1.1111..11   -269        805 11..1..1.1   -219        855 11,1.1,111   -169
756 1.1111.1..   -268        806 11..1..11.   -218        856 11,1.11,,,   -168
757 1.1111.1.1   -267        807 11..1..111   -217        857 11,1.11,,1   -167
758 1.1111.11.   -266        808 11..1.1...   -216        858 11,1.11,1,   -166
759 1.1111.111   -265        809 11..1.1..1   -215        859 11,1.11,11   -165
760 1.11111...   -264        810 11..1.1.1.   -214        860 11,1.111,,   -164
761 1.11111..1   -263        811 11..1.1.11   -213        861 11,1.111,1   -163
762 1.11111.1.   -262        812 11..1.11..   -212        862 11,1.1111,   -162
763 1.11111.11   -261        813 11..1.11.1   -211        863 11,1.11111   -161
764 1.111111..   -260        814 11..1.111.   -210        864 11,11,,,,,   -160
765 1.111111.1   -259        815 11..1.1111   -209        865 11,11,,,,1   -159
766 1.1111111.   -258        816 11..11....   -208        866 11,11,,,1,   -158
767 1.11111111   -257        817 11..11...1   -207        867 11,11,,,11   -157
768 11........   -256        818 11..11..1.   -206        868 11,11,,1,,   -156
769 11.......1   -255        819 11..11..11   -205        869 11,11,,1,1   -155
770 11......1.   -254        820 11..11.1..   -204        870 11,11,,11,   -154
771 11......11   -253        821 11..11.1.1   -203        871 11,11,,111   -153
772 11.....1..   -252        822 11..11.11.   -202        872 11,11,1,,,   -152
773 11.....1.1   -251        823 11..11.111   -201        873 11,11,1,,1   -151
774 11.....11.   -250        824 11..111...   -200        874 11,11,1,1,   -150
775 11.....111   -249        825 11..111..1   -199        875 11,11,1,11   -149
776 11....1...   -248        826 11..111.1.   -198        876 11,11,11,,   -148
777 11....1..1   -247        827 11..111.11   -197        877 11,11,11,1   -147
778 11....1.1.   -246        828 11..1111..   -196        878 11,11,111,   -146
779 11....1.11   -245        829 11..1111.1   -195        879 11,11,1111   -145
780 11....11..   -244        830 11..11111.   -194        880 11,111,,,,   -144
781 11....11.1   -243        831 11..111111   -193        881 11,111,,,1   -143
782 11....111.   -242        832 11.1......   -192        882 11,111,,1,   -142
783 11....1111   -241        833 11.1.....1   -191        883 11,111,,11   -141
784 11...1....   -240        834 11.1....1.   -190        884 11,111,1,,   -140
785 11...1...1   -239        835 11.1....11   -189        885 11,111,1,1   -139
786 11...1..1.   -238        836 11.1...1..   -188        886 11,111,11,   -138
787 11...1..11   -237        837 11.1...1.1   -187        887 11,111,111   -137
788 11...1.1..   -236        838 11.1...11.   -186        888 11,1111,,,   -136
789 11...1.1.1   -235        839 11.1...111   -185        889 11,1111,,1   -135
790 11...1.11.   -234        840 11.1..1...   -184        890 11,1111,1,   -134
791 11...1.111   -233        841 11.1..1..1   -183        891 11,1111,11   -133
792 11...11...   -232        842 11.1..1.1.   -182        892 11,11111,,   -132
793 11...11..1   -231        843 11.1..1.11   -181        893 11,11111,1   -131
794 11...11.1.   -230        844 11.1..11..   -180        894 11,111111,   -130
795 11...11.11   -229        845 11.1..11.1   -179        895 11,1111111   -129
796 11...111..   -228        846 11.1..111.   -178        896 111,,,,,,,   -128
797 11...111.1   -227        847 11.1..1111   -177        897 111,,,,,,1   -127
798 11...1111.   -226        848 11.1.1....   -176        898 111,,,,,1,   -126
799 11...11111   -225        849 11.1.1...1   -175        899 111,,,,,11   -125
```

```
900 111....1..    -124
901 111....1.1    -123
902 111....11.    -122
903 111....111    -121
904 111...1...    -120
905 111...1..1    -119
906 111...1.1.    -118
907 111...1.11    -117
908 111...11..    -116
909 111...11.1    -115
910 111...111.    -114
911 111...1111    -113
912 111..1....    -112
913 111..1...1    -111
914 111..1..1.    -110
915 111..1..11    -109
916 111..1.1..    -108
917 111..1.1.1    -107
918 111..1.11.    -106
919 111..1.111    -105
920 111..11...    -104
921 111..11..1    -103
922 111..11.1.    -102
923 111..11.11    -101
924 111..111..    -100
925 111..111.1     -99
926 111..1111.     -98
927 111..11111     -97
928 111.1.....     -96
929 111.1....1     -95
930 111.1...1.     -94
931 111.1...11     -93
932 111.1..1..     -92
933 111.1..1.1     -91
934 111.1..11.     -90
935 111.1..111     -89
936 111.1.1...     -88
937 111.1.1..1     -87
938 111.1.1.1.     -86
939 111.1.1.11     -85
940 111.1.11..     -84
941 111.1.11.1     -83
942 111.1.111.     -82
943 111.1.1111     -81
944 111.11....     -80
945 111.11...1     -79
946 111.11..1.     -78
947 111.11..11     -77
948 111.11.1..     -76
949 111.11.1.1     -75

950 111.11.11.     -74
951 111.11.111     -73
952 111.111...     -72
953 111.111..1     -71
954 111.111.1.     -70
955 111.111.11     -69
956 111.1111..     -68
957 111.1111.1     -67
958 111.11111.     -66
959 111.111111     -65
960 1111......     -64
961 1111.....1     -63
962 1111....1.     -62
963 1111....11     -61
964 1111...1..     -60
965 1111...1.1     -59
966 1111...11.     -58
967 1111...111     -57
968 1111..1...     -56
969 1111..1..1     -55
970 1111..1.1.     -54
971 1111..1.11     -53
972 1111..11..     -52
973 1111..11.1     -51
974 1111..111.     -50
975 1111..1111     -49
976 1111.1....     -48
977 1111.1...1     -47
978 1111.1..1.     -46
979 1111.1..11     -45
980 1111.1.1..     -44
981 1111.1.1.1     -43
982 1111.1.11.     -42
983 1111.1.111     -41
984 1111.11...     -40
985 1111.11..1     -39
986 1111.11.1.     -38
987 1111.11.11     -37
988 1111.111..     -36
989 1111.111.1     -35
990 1111.1111.     -34
991 1111.11111     -33
992 11111.....     -32
993 11111....1     -31
994 11111...1.     -30
995 11111...11     -29
996 11111..1..     -28
997 11111..1.1     -27
998 11111..11.     -26
999 11111..111     -25

1000 11111.1...    -24
1001 11111.1..1    -23
1002 11111.1.1.    -22
1003 11111.1.11    -21
1004 11111.11..    -20
1005 11111.11.1    -19
1006 11111.111.    -18
1007 11111.1111    -17
1008 111111....    -16
1009 111111...1    -15
1010 111111..1.    -14
1011 111111..11    -13
1012 111111.1..    -12
1013 111111.1.1    -11
1014 111111.11.    -10
1015 111111.111     -9
1016 1111111...     -8
1017 1111111..1     -7
1018 1111111.1.     -6
1019 1111111.11     -5
1020 11111111..     -4
1021 11111111.1     -3
1022 111111111.     -2
1023 1111111111     -1
1024 ..........      0
1025 .........1      1
1026 ........1.      2
1027 ........11      3
1028 .......1..      4
1029 .......1.1      5
1030 .......11.      6
1031 .......111      7
1032 ......1...      8
1033 ......1..1      9
1034 ......1.1.     10
1035 ......1.11     11
1036 ......11..     12
1037 ......11.1     13
1038 ......111.     14
1039 ......1111     15
1040 .....1....     16
1041 .....1...1     17
1042 .....1..1.     18
1043 .....1..11     19
1044 .....1.1..     20
1045 .....1.1.1     21
1046 .....1.11.     22
1047 .....1.111     23
1048 .....11...     24
1049 .....11..1     25
```