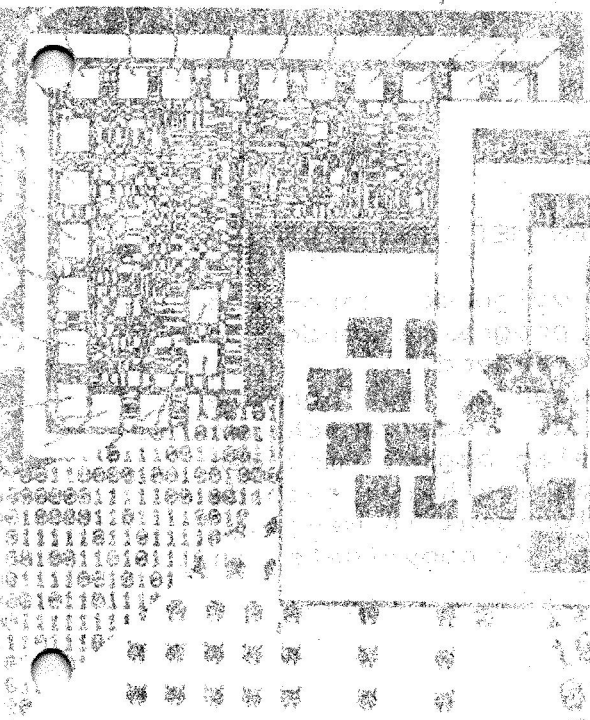
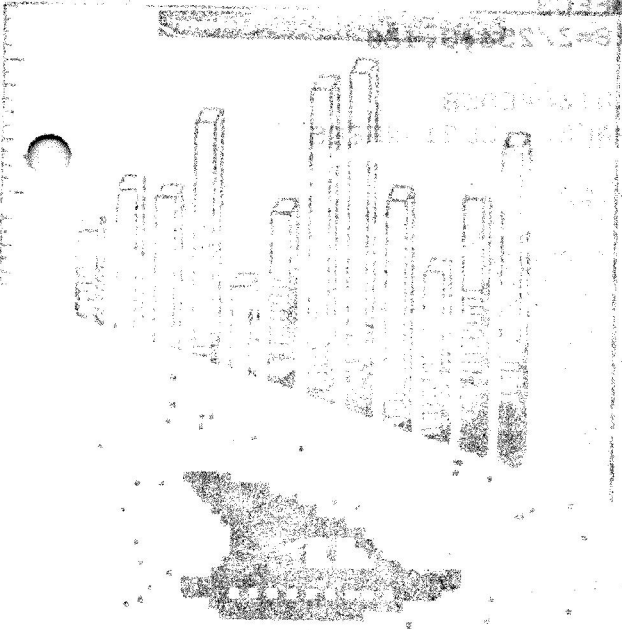
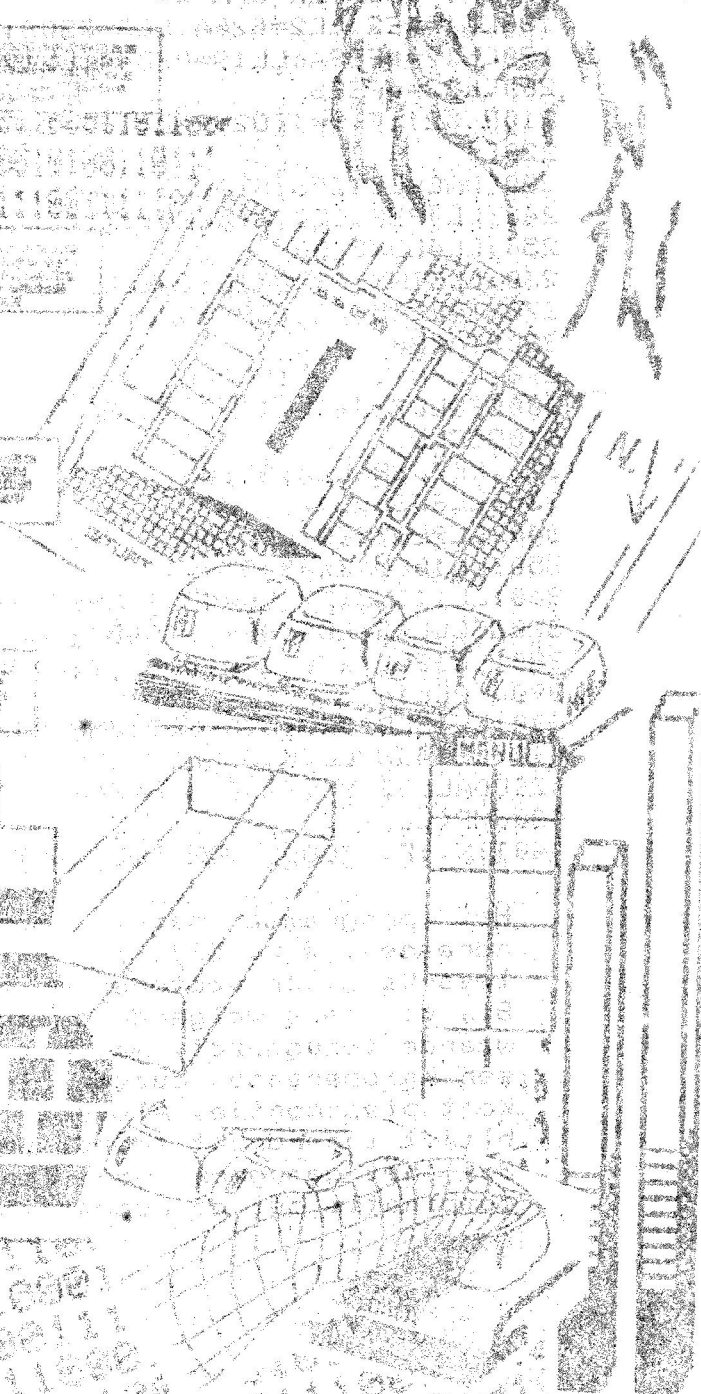
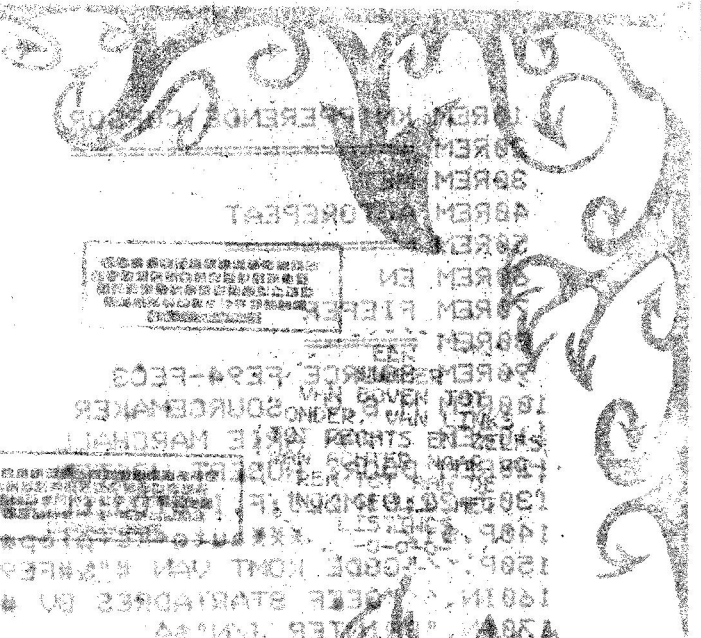


2e JAARGANG  
BRONSGROEN  
EIKELTJE no. 3

BRONSGROEN  
EIKELTJE  
no. 3



1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900



1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900

```

10REM KNIPPERENDE CURSOR
20REM =====
30REM MET
40REM AUTOREPEAT
50REM =====
60REM EN
70REM PIEPER
80REM =====
90REM SOURCE FE94-FEC3
100REM M.B.V. SOURCEMAKER
110REM VAN ARIE MARCHALL
120REM DOOR: HUBERT VOGTEN
130J=22;DIMLLJ;F.I=0TOJ;LLI=#FFF;N.;DIMA4;@=0
140P.$12"   ***auto+Kc+pieper***"
150P.' "CODE KOMT VAN "&#FE94;P." - #"&#FEC3
160IN.' "GEEF STARTADRES BV #3000 "Z;IF?18=Z/256;G.160
170IN."PRINTER J/N"$A
180LL1=#E2;LL2=#20A;LL3=#20B;LL15=#B002;LL16=#C55B
190LL17=#FB8A;LL18=#FD44;LL19=#FE66;LL20=#FE71;LL21=#FEC5
200LL22=#FEE3
210P.$21;F.I=1TO2;P=Z;IFI=2;P.$6;IF?A=74P.$2
220I
230LDA@LL0/256;STALL2;LDA@LL0/256;STALL3;JMPLL16
240:LL0PHP;CLD;STX#E4;STY#E5;LDY@#00
250:LL4LDX@#00
260:LL5INX;CPX@#1E;BNELL5;LDALL15;EOR@#04
270STALL15;INY;CPY@#3C;BNELL4;LDA@#00
280STA#B0
290:LL6BITLL15;BVCLL9;INC#B0;LDA#B0;CMP@#20
300BEQLL7;JSRLL19;JSRLL20;CPY#B1;BEQLL8
310BCCLL6
320:LL7JSRLL20;STY#B1
330:LL8JSRLL17
340:LL9LDY#E0;LDA(#DE),Y;STA#B3
350:LL10LDA@#00;STA#B2
360:LL11INC#B2;LDA#B2;CMP@#FF;BEQLL13;JSRLL20
370BCCLL12;LDA@#FF;STA#B1;JMPLL11
380:LL12LDY#E0;LDA(#DE),Y;CMP#B3;BEQLL14;JSRLL18
390JMPLL14
400:LL13LDY#E0;JSRLL18;JMPLL10
410:LL14JSRLL20;BCSL19;TYA;LDX@#17;JSRLL21
420LDALL22,X;STA#E2;LDA@#FD;STA#E3;TYA
430JMP(LL1)
440J;N.;P.' "CODE VAN "&Z" TOT "&P'/$3"STARTEN MET LINK #"&Z';E.

```

Het programma maakt gebruik van de volgende zero-page adressen: #B0, #B1, #B2 en #B4. Door het programma worden de lees-karakter vectoren op #20A en #20B veranderd. Bij de eerstvolgende break worden ze weer op hun normale waarde teruggezet. Na het aanklinken van de code verschijnt een knipperende cursor. Iedere toetsaanslag begint met een kort pieptoontje. Als een toets langer dan een halve seconde blijft ingedrukt treedt bovendien de autorepeat in werking. Bij het debuggen van een programma is de copy-toets in combinatie met de repeat-toets erg handig.

\*\*\*\*\*  
BERICHT VAN HET DRUKWERKARCHIEF B. TOSSAINT  
\*\*\*\*\*

### 1 BBC-literatuur

Zoals op de jaarvergadering vastgesteld, zal in het drukwerkarchief ook materiaal voor BBC verzameld worden; de eerste 30 items zijn inmiddels al verzameld op basis van de in ons bezit zijnde tijdschriften (voornamelijk Your Computer), samengevoegd in een afzonderlijke map (P), die op gelijke wijze is georganiseerd als het ATOM-archief, nl. hardware, systeemsoftware, applicatiesoftware.

Nu waren we reeds bezig de activiteiten van het archief te spreiden over meerdere mensen; in dit verband lijkt het voor de hand liggend dat een der BBC-bezitters tot het drukwerkarchief-team toetreedt om specifiek dat deel te verzorgen.

WIE.

### 2 Beschrijving operating system ATOM

H. Reinders van de regio Noord heeft de moeite genomen een zeer uitgebreide beschrijving en verklaring op te stellen van de routines van het operating system, in concreto de routines opgeborgen in het grote BK rom, ofwel de adressen Cxxx en Fxxx, of voor de nog niet ingewijde, het eigenlijke interne besturingsprogramma van de computer zelf.

Deze toelichting is in feite alleen interessant voor diegenen die in assembler werken en gebruik willen maken van de onderdelen van de routines. De totale omvang bedraagt 70 pagina's; kosten Fl.7,- per exemplaar. Gezien de omvang en de reeds ontvangen bestellingen is het de bedoeling dit geschrift 'extern' te laten drukken/copieren. Gaarne opgave VOOR 5 mei, op clubavond, of telefonisch bij een der leden van het drukwerkarchief (archieffnummer F017).

### 3 Inzamelen literatuur

In aansluiting op de jaarvergadering zouden wij nogmaals de aandacht willen vragen voor afspraken tav. literatuurinzameling. Het zeer grote aantal (en dure) computertijdschriften maakt het onmogelijk deze allemaal te verwerken. Daarom willen wij een overzicht krijgen van:

- diegenen die geabonneerd zijn op tijdschriften
  - diegenen die bepaalde tijdschriften screenen en kopen indien er voor henzelf iets belangrijks in te vinden is
- teneinde afspraken te maken tav. het beschikbaar stellen van artikelen die van belang zouden kunnen zijn voor de club.

Jansen vMeegen Tossaint

\*\*\*\*\*  
 BEGINNERSHOEKJE E. SANDERS  
 \*\*\*\*\*

Speciaal voor de beginners onder ons en voor degenen die moeite hebben met het Engels (en dat zijn er meer dan we denken) zal ik met ingang van dit Eikeltje een rubriek starten met de vertaling van het handboek.

Niet klakkeloos de vertaling maar daar waar nodig en mogelijk zal de nodige uitleg en commentaar geleverd worden.

Ik begin met die hoofdstukken waar de meeste moeite aan bestaat. Ze komen echter allemaal aan de beurt.

Eerst even een toelichting op het hierna te beginnen hoofdstuk 19. Dit gaat over het COS (Cassette Operating System).

Omdat SAVE en LOAD niet tot het COS behoren maar tot de normale(?) ATOM-basic zul je hierover weinig in dit hoofdstuk terugvinden, het komt echter allemaal nog aan de orde. Dan zullen we nu maar beginnen.

### CASSETTE OPERATING SYSTEM

Het Cassette Operating System, hierna te noemen: C.O.S. Leadt data van band en schrijft data weg naar band door gebruik te maken van de Computer Users Tape Standard (CUTS), welke ook bekend staat als de Kansas City Standard.

De data is als audiogeluid op tape opgeslagen. Een logische 0 bestaat uit 4 perioden van 1,2 Khz. en een logische 1 bestaat uit 8 perioden van 2,4 Khz. Elk databyte wordt vooraf gegaan door een logische 0; het startbit en wordt afgesloten door een logische 1; het stopbit.

Elk bit duurt 3,33 msec. hetgeen resulteert in een verwerkingsnelheid van 300 bits/sec.

#### 19.1 Benoemde files (dus files gesaved met een naam)

Benoemde files worden opgeslagen in een aantal blokken, elk met 256 byte's of (voor de laatste blok in veel gevallen) minder, inclusief de "checksum".

De checksum is een controlegetal, dat nodig is om te kijken of er tijdens het laden of saven een of meerdere byte's verminkt zijn. Ieder blok wordt aangeduidt door middel van een "naamlabel"; hierin wordt vermeld: de naam van het programma, het startadres om het blok te laden, het executieadres van het programma en het aantal byte's in dat blok minus 1.

#### 19.2 Niet benoemde files (dus files zonder naam)

Niet benoemde files worden opgeslagen als een groot blok. Een checksum is hierbij niet aanwezig.

De blokaanduiding bevat nu de volgende gegevens: een 2-bytes startadres, een 2-bytes eindadres en het aantal bytes; dwz het eindadres - startadres = aantal data bytes.

Een onbenoemde file kan dus geen naam hebben (bij gebruik van \*LOAD en \*SAVE) of het zou een naam moeten zijn van 0 posities aangeduid met "". Onbenoemde files kunnen overal gebruikt worden waar benoemde files zijn toegestaan.

#### 19.3 Beschrijving van de kommando's

Alle COS kommando's beginnen met een ster (\*), dit om ze te kunnen onderscheiden van BASIC kommando's.

Let dus op het verschil tussen SAVE en \*SAVE, LOAD en \*LOAD.

SAVE schrijft de files weg vanaf de textspace waar U zich op dat moment bevindt. Er wordt geen startadres opgegeven. Het

executieadres wordt automatisch naar #C2B2 gezet.

SAVE kan alleen gebruikt worden voor het wegschrijven van

BASIC-programma's. Waarom? omdat bij SAVE de computer automatisch

op zoek gaat naar de TOP en aangezien machinetaal geen TOP heeft

kan de computer deze TOP niet vinden en komt na het laden van een

machinetaalprogramma meestal ook niet terug met de prompt (>).

\*SAVE schrijft een stuk geheugen weg waarvan het start- en het eindadres (+1) moeten worden opgegeven.

LOAD laadt de files naar de textspace waar U zich op dat moment bevindt.

\*LOAD laadt een file naar een vast adres (het adres wat bij \*SAVE meegegeven is geworden) of naar een adres wat opgegeven is in het kommando. (zie verderop)

\*CAT Catalogiseer de tape

Het \*CAT kommando geeft een opsomming van alle programma's die op de band staan. Iedere blok van een met naam gesavede file zal als volgt in de catalog verschijnen:

FILENAAM SSSS EEEE NNNN BB

Hierin is FILENAAM de naam van de file.

SSSS het startadres van het blok

EEEE het executieadres van de file (wordt gebruikt bij RUN)

NNNN het bloknummer, beginnend bij 0

BB het aantal bytes in een blok min 1.

Alle getallen zijn natuurlijk hexadecimaal.

Files die zonder naam gesaved zijn verschijnen als volgt in de catalog:

SSSS LLLL

waarin SSSS het startadres en LLLL het laatste adres, plus 1 is. Alles weer hexadecimaal.

\*LOAD laad file

Om een file te laden luidt het volledige kommando:

\*LOAD"FILENAAM"XXXX

waarin XXXX het hexadecimale adres is waar de file naar toe geladen moet worden. Als je XXXX weg laat wordt de file geladen

vanaf het adres waarvan het oorspronkelijk gekomen is. Als je na het voorgaande op RETURN drukt zal het systeem antwoorden met:

PLAY TAPE

De cassetterecorder wordt nu gestart, waarna je op de spatiebalk drukt om aan te geven dat dit gebeurd is.

Het COS zal de namen laten zien van iedere file die hij tegenkomt op de band voordat de opgegeven file is gevonden. Als de opgegeven file is gevonden zal deze worden geladen en als

alles geladen is zal ook weer de ">" verschijnen. Als je een file wilt laden en je bent reeds voorbij het eerste blok dan zal het COS de volgende boodschap laten zien:

REWIND TAPE

Spoel de band dan terug tot voor het eerste blok en druk weer op de spatiebalk waarop het COS zal antwoorden:

PLAY TAPE

en het laadproces kan worden herhaald.

Om een file zonder naam te laden luidt het volledige kommando:

```
*LOAD**XXXX of
*LOAD XXXX
```

waarin XXXX wederom het, hexadecimale, startadres naar keuze is. Omdat de computer niet naar een naam hoeft te zoeken kan men het beste de spatiebalk indrukken gedurende de "high-tone leader", dit is de hoge toon die aan de file voorafgaat. Door tijdens de high-tone leader de spatiebalk in te drukken zal alleen die file geladen worden die de computer hierna tegenkomt. Zoals gezegd bestaat een file zonder naam uit een enkel (groot) blok en er is geen checksum controle. Het is echter de snelste manier om data of programma's te laden of te save. Men kan alleen niet met een \*CAT zien welk programma het is.

#### CTRL en SHIFT

Het gebruik van deze toetsen tijdens het laden en \*CAT:

CTRL zal zorgen dat de ATDM terugkeert naar de ">" prompt. Als men CTRL indrukt tijdens het laden zal er een error verschijnen om aan te geven dat dat deel van de file verloren is gegaan.

SHIFT ingedrukt zal het zoeken naar de high-tone leader achterwegelaten en kan dus worden gebruikt om files te laden en te catalogiseren met een zeer korte high-tone leader.

Denk eraan dat er geen manier is om uit een SAVE of een \*SAVE te komen. Dit kan alleen maar met een BREAK.

#### \*SAVE save file

Om een file met naam op de tape te save luidt het volledige kommando:

```
*SAVE"FILENAAM"SSSS LLLL EEEE
```

waarin "FILENAAM" de naam is van de file, deze naam mag maximaal 15 karakters bevatten. Dit kan echter per computer variëren.

SSSS is het startadres

LLLL is het eindadres +1

EEEE is het executieadres naar keuze

Het executieadres wordt door de computer gebruikt bij het RUN kommando. Als dit weg wordt gelaten krijgt het executieadres automatisch de waarde van het startadres.

Na op de return toets te hebben gedrukt antwoord het COS met:

#### RECORD TAPE

De cassetterecorder wordt nu gestart in de opname stand, waarna de spatiebalk wordt ingedrukt om aan te geven dat de recorder loopt.

Als de computer eenmaal aan het save is kan dit alleen nog maar onderbroken worden met een BREAK.

Om een file zonder naam te kunnen save luidt het volledige kommando:

```
*SAVE""SSSS LLLL of
```

```
*SAVE SSSS LLLL
```

waarin SSSS weer het startadres en LLLL het eindadres is. De data wordt nu gesaved als een groot blok. (zonder checksum)

Ook hierbij kan naar believe het executieadres worden ingevuld of worden weggelaten.

Tot zover de uitleg van het COS.

De volgende keer gaan we verder met de andere kommando's van het COS en er zal aandacht worden besteed aan enkele speciale wetenswaardigheden van het COS.

Mochten er nog vragen zijn dan hoor ik die graag.

\*\*\*\*\*  
EPROM PROGRAMMER SOFT W. SALDEN  
\*\*\*\*\*

```
10 REM EPROM PROGRAMMER
30 E=#80;B=#81;C=#82;D=#83
40 DIMWW40,II5;F.F=0TO45;WW(F)=-1;N.
50 P.$21;F.F=1TO2;P=#9400;I
60:II5 JSR#F7D1;I;$P=" switch to 5 volt ";?P=10
70 P=P+LENP;I;NOP;JSR#FC76;JSR#FFED
80:II0 LDA@0;STA B;STA E;STA C;STA#B80E;LDA#84;STA D
90 LDA@#CC;STA#B80C;LDY@0;RTS
100:II1 LDA@#FF;STA#B802;STA#B803
110 LDA B;STA#B800;LDA E;STA#B801
120 LDA@#CE;STA#B80C;LDA@#CC;STA#B80C;RTS
130:II4 LDA@#CC;STA#B80C
140:II2 INY;INC E;BNEP+6;INC D;INC B
150 LDA B;CMP@16;RTS
160:II3 LDA@0;STA#B802;LDA@#EC;STA#B80C;LDA#B800;RTS
169:WW0 JSR#AFDB
170 LDA@85;STA#208
180 JSR#FD69;LDY@7;JSR#F99A;JSR#F7D1;I;$P="2532-prog A8BK-ii "
190 GOS.a;P?-2=10
200 $P=" menu ";GOS.a;P?-3=10
210 $P="1 PROGRAM ";GOS.a
220 $P="2 VERIFY ";GOS.a
230 $P="3 COPY ";GOS.a
240 $P="4 LIST ";GOS.a
250 $P="5 EMPTY ";GOS.a
260 $P="6 CHECKSUM ";GOS.a
270 $P="7 STOP ";GOS.a;P?-3=10
280 $P=" UW KEUS? ";P=P+LENP
290I;NOP;JSR#FFE3;PHA;JSR#FD69;PLA;CMP@49;BNEP+4;BEQWW1
300 CMP@50;BNEP+5;JMPWW10;CMP@51;BNEP+5;JMPWW20
310 CMP@52;BNEP+5;JMPWW25;CMP@53;BNEP+5;JMPWW32
320 CMP@54;BNEP+5;JMPWW36;CMP@55;BNEP+5;JMPWW39
330 CMP@27;BNEP+5;JMPWW39;JMPWW0
340\PROGRAM
350:WW1 JSR#F7D1;I;$P="switch to 25 volt ";GOS.a;P?-3=10
360I;NOP;JSR#FFE3;JSR#F7D1;I
370 $P="PROGRAMMING EPROM ";GOS.a;P?-3=10;I;NOP;JSRII0
380:WW2 JSR II1;LDA(C),Y;STA#B800;LDA@#EC;STA#B80C
390 LDX@3;JSR#FB83;JSRII4;BEQWW3;BNEWW2
400:WW3 JSR#FD1A;JSR#FD1A;JSR#FD1A;JSRII5;JMPWW0
410\VERIFY
420:WW10 LDY@5;JSR#F99A;JSR#F7D1;I;$P="verifying ";GOS.a
430 P?-3=10;I;NOP;JSRII5
440:WW11 JSRII1;JSRII3;CMP(C),Y;BNEWW15
450:WW12 JSRII4;BEQWW17;BNEWW11
460:WW15 LDA#B001;CMP@191;BNEWW16;JMPWW0
470:WW16 STY#85;JSR#F7D1;I;$P="NO MATCHING:";P=P+LENP;I;LDY#85
480 LDX@E;JSR#F7F1;LDA@58;JSR#FFF4;LDA#B800
490 JSR#F7FA;LDA(C),Y;JSR#F802;JSR#FFED;JMPWW12
```

```

500:WW17 JSR#F7D1; J; $P="OK ! "; P=P+LENP; C; NOP
510:WW18 JSR#FFE3; JMPWWO
520\COPY
530:WW20 JSRII5
540:WW21 JSRII1; JSRII3
550 STA(C), Y; JSRII4; BEQWW22; BNEWW21
560:WW22 JSR#F7D1; J; $P="COPIED ! "; P=P+LENP; C; NOP; JMPWW18
570\LIST
580:WW25 JSRII5
590:WW26 LDA#B001; CMP#191; BNEWW27; JMPWWO
600:WW27 LDX#E; JSR#F7F1; LDA#58; JSR#FFF4; JSR#F7FD
610 JSRII1; JSRII3; BMIWW28; CMP#20; BCCWW28
620 JSR#FFF4; JSR#F7FD; JMPWW29
630:WW28 JSR#F7FD; JSR#F7FD
640:WW29 LDA#B800; JSR#F802; JSR#FFED; JSRII4; BNEWW28; JMPWW18
650\EMPTY
660:WW32 JSRII5
670:WW33 JSRII1; JSRII3; CMP#FF; BNEWW35; JSRII4; BNEWW33
680:WW34 JSR#F7D1; J; $P="EMPTY ! "; P=P+LENP; C; NOP; JMPWW18
690:WW35 JSR#F7D1; J; $P="NOT "; P=P+LENP; C; NOP; BNEWW34
700\CHECKSUM
710:WW36 JSRII5; LDA#0; STA#AA; STA#AB; STA#AC
720:WW38 JSRII1; JSRII3; CLC; ADC#AA; STA#AA; BCSWW37; BCCWW30
730:WW37 INC#AB; BNEWW30; INC#AC
740:WW30 JSRII4; BEQWW31; BNEWW39
750:WW31 JSR#F7D1; J; $P="CHECKSUM = "; P=P+LENP
760 C; NOP; LDA#AC; BEQP+5; JSR#F802; LDX#AA; JSR#F7F1
770 JMPWW18.
780:WW39 JSR#FFED; JMP#C558
790J; N.; P. $6; E.
800aP=P+LENP+1; P?-2=10; R.

```

\*\*\*\*\*  
 BERICHT VOOR BEGINNERS EN GEINTERESSEERDEN  
 \*\*\*\*\*  
 Vooral voor de beginners lijkt het zinvol de aandacht te vestigen op enkele televisieprogramma's, waarin de grondslagen van de computer-techniek op een goede wijze aan de man gebracht worden i.c.:  
 BRT-1 Za. 15.00 uur en  
 Do. 19.00 uur "het ABC v.d. Computer"  
 BRT-2 Wo. 19.00 uur "CHIPS" om de 14 dgn.

Bruno Tossaint

410/VERIFY  
 420 P?-2=10; R.  
 430 JSRII5  
 440:WW11 JSRII1; JSRII3; CMP#FF; BNEWW35; JSRII4; BNEWW33  
 450:WW12 JSRII1; JSRII3; CLC; ADC#AA; STA#AA; BCSWW37; BCCWW30  
 460:WW13 JSRII5; LDA#0; STA#AA; STA#AB; STA#AC  
 470:WW14 JSR#F7D1; J; \$P="EMPTY ! "; P=P+LENP; C; NOP; JMPWW18  
 480 LDX#E; JSR#F7F1; LDA#58; JSR#FFF4; JSR#F7FD  
 490 JSR#F7FD; JSR#F7FD



\*\*\*\*\*  
SCHIJVEN-ARCHIEF W. ERNST  
\*\*\*\*\*

Gezien het aantal floppy-discs welke in de club aanwezig zijn, leek het bij de afgelopen bestuursvergadering wenselijk om een software-archief op schijf op te bouwen. Aangezien ik de mogelijkheid heb tot gemakkelijk kopiëren van schijven via een dubbele drive, heb ik deze taak op mij genomen.

De in dit archief opgeslagen software zal gedeeltelijk parallel lopen met het bandjesarchief en gedeeltelijk andere software bevatten.

Het archief treedt in werking met het uitkomen van dit blaadje. Dus vanaf nu kan men hiervoor bij mij terecht.

Het archief zal als volgt gaan functioneren:

- bestellingen worden gedaan op de regio-avond bij vooruitbetaling van de kosten ad Fl.12,50; hierin is inbegrepen een schijf van goede kwaliteit (verbatim/scotch) en de zich daarop bevindende soft.
- deze bestellingen kunnen op de volgende regio-avond afgehaald worden, of tussendoor met de post verstuurd tegen vergoeding van de porto- en verpakkingskosten, welke ca. Fl.2,50 zullen bedragen.

Dokumentatie, voor zover noodzakelijk en/of aanwezig, bevindt zich in het drukwerkarchief.

We hopen hiermee alle floppy-bezitters een plezier te kunnen doen.

Namens het bestuur  
W. Ernst

\*\*\*\*\*  
READ DATA RESTORE W. ERNST  
\*\*\*\*\*

Er zijn mij nogal wat vragen gesteld over de READ DATA RESTORE statements. Daarom deze poging hierin enige opheldering te brengen.

READ is het Engelse woord voor lezen of inlezen. Met deze opdracht kunnen we aan een variabele een waarde toekennen, of indien het een stringvariabele betreft een string toewijzen. Bij een stringvariabele moet men echter deze wel dimensioneren; bv DIM A(15), waarmee bepaald wordt dat \$A 15 karakters mag bevatten.

DATA is het Engelse woord voor gegevens. Achter dit woord staan de getallen of de strings die met behulp van de READ opdracht aan de betreffende variabelen toegekend worden.

RESTORE is het Engelse woord voor herstellen. Om bij te houden hoeveel DATA er ingelezen is, wordt een teller bijgehouden. Wil men nu de waarden opnieuw inlezen, dan geeft men de opdracht RESTORE. Hierdoor wordt de teller op het eerste DATA-item gezet en kunnen deze opnieuw worden ingelezen.

Uitgaande van de JOSBOX zal ik nu enkele voorbeelden geven met enige toelichting.

#### Voorbeeld 1

Het inlezen van getallen.

```
10 RESTORE
20 READ A,B,C,D,E
30 PRINT A,B,C,D,E
40 DATA 10,45,56,1,90
50 END
```

Hierin krijgt A de waarde 10, B de waarde 45, enz.

#### Voorbeeld 2

Het inlezen van strings.

```
10 RESTORE
20 DIM A(10),B(10),C(10),D(10),E(10)
30 READ $A,$B,$C,$D,$E
40 PRINT $A,$B,$C,$D,$E
50 DATA "ACORN ATOM","DEMO READ","DATA EN"
60 DATA "RESTORE","OPDRACHTEN"
70 END
```

Hierin wordt \$A ACORN ATOM, \$B wordt DEMO READ, enz.

#### Voorbeeld 3

Het inlezen van array's.

```
10 RESTORE
20 DIM AA(10)
30 FOR N=0 TO 10
40 READ AA(N)
50 NEXT N
60 FOR N=0 TO 10
70 PRINT AA(N)
80 NEXT N
90 DATA 10,40,50,25,60,78,112,3456,456,999,123456
100 END
```

Hierin krijgt AA(0) de waarde 10, AA(1) de waarde 40, AA(2) de waarde 50, enz.

Men kan ook een string-array inlezen mits deze correct is gedefinieerd.

Bijvoorbeeld:

```
10 RESTORE
20 DIM AA(5)
30 FOR N=0 TO 5
40 DIM X(10)
```

```
50 AA(N)=X
60 NEXT N
70 FOR N=0 TO 5
80 READ $AA(N)
90 NEXT N
100 .....
110 .....
120 DATA "JANSSEN", "WOONT", "OP", "ADRES", "TE", "WOONPLAATS"
130 END
```

Betreffende de RESTORE opdracht uit de JOSBOX zijn nog de volgende opmerkingen te maken:

- alleen RESTORE; dan wordt de teller (pointer) gezet op het eerste DATA-item.
- RESTORE 100; dan wordt de teller gezet op het eerste DATA-item op regel 100.
- RESTORE a; dan wordt de teller gezet op het eerste DATA-item van de regel beginnend met label 'a'.
- RESTORE (5\*I+6); dan wordt de teller gezet op het eerste DATA-item van het berekende regelnummer, welke afhankelijk is van de variabele 'I'.

Opmerking 1:

Omdat bij de ATOM de teller (pointer) bij het aanzetten van het toestel een willekeurige waarde heeft, moet bij gebruik van een READ DATA opdracht aan het begin van het programma eerst een RESTORE opdracht gegeven worden, om deze pointer goed te zetten.

Opmerking 2:

Men dient er voor te zorgen dat voor elke READ opdracht er ook een DATA-item beschikbaar is. Dat wil zeggen, wil men met READ 10 getallen inlezen, er ook 10 getallen achter DATA moeten staan. Is dit niet het geval, dan volgt een error-melding.

Ik hoop hiermee voor enkele of meerdere mensen enige duidelijkheid te hebben gebracht in de READ DATA RESTORE constructie.

Wim Ernst

```
*****
GRAPJE VAN HOENEN (knipoog)
*****
```

toets maar eens in.

```
10 REM **ANIMATIE**
20 TKON; REM TOOLBOX ON
30 GOSUB b; REM WIS SCHERM
40 RESTORE
45 READ D,R
50 IF D=99 GOTO 100
60 GOSUB f; REM POSITIE CURSOR
```

```

70 GOSUB c; REM WIS CURSOR
80 PRINT "X"
90 GOTO 45
100 D=4;R=6; REM KNIPPEREN
110 GOSUB c
120 FOR I=1 TO 10
130 GOSUB f; PRINT "—"; GOSUB d
140 GOSUB g; PRINT "X"; GOSUB d
150 NEXT I
160 GOSUB e; REM HOME
200 PRINT #12; RETURN
400 FOR J=1 TO 10; WAIT; NEXT J
500 CURSOR R,D; RETURN
1000 DATA 1,3,1,4,1,5,1,6,1,7
1010 DATA 2,2,2,8
1020 DATA 3,1,3,3,3,4,3,6,3,7,3,9
1030 DATA 4,0,4,1,4,3,4,4,4,6,4,7,4,9,4,10
1040 DATA 5,0,5,1,5,9,5,10
1050 DATA 6,0,6,1,6,5,6,7,6,10
1060 DATA 7,0,7,1,7,8,7,9,7,10
1070 DATA 8,0,8,1,8,5,8,6,8,9,8,10
1080 DATA 9,0,9,1,9,5,9,6,9,10
1090 DATA 10,0,10,1,10,5,10,6,10,9,10,10
1100 DATA 99,99
1200 REM HOENEN
1210 REM BRUNNEN
1220 REM BESSEVEN
1230 REM COMPUTERS
1240 REM R. V. D. BONGEN

```

Ik hoop hiermee voor enkele andere mensen enige duidelijkheid te hebben. Het programma is een constructie van INFOMASTER B. TOSSAINT.

Nieuwe ontwikkelingen aan en wetenschappelijke van het Infomaster-terrein

Naar aanleiding van vragen van verschillende gebruikers heeft Marchal weer in zijn toverdoos gedoken en heeft enkele aanvullingen voor het infomaster programma geschreven.

- 1) overbrengen van een bestand, aangemaakt met INFO2(L) naar Infomaster
- 2) formatteren (=corrigeren) van de scherm-afbeelding
- 3) het OPTELLEN van de waarde die een record heeft van de aanwezige records
- 4) enkele hulpmiddelen

1 Overbrengen van databestanden van INFO2 naar Infomaster  
 Vroeger gebruikers van eerdere versies van Infomaster, die INFO2 bestanden hebben, kunnen deze nu overbrengen naar Infomaster.

of INFOB2L, hebben wellicht de behoefte het databestand over te brengen naar Infomaster, zonder dit opnieuw helemaal met de hand te moeten invoeren.

Doordat in InfoB2(L) VASTE geheugengebieden werden gebruikt voor de 'record'- en 'ascii'-file, terwijl dit in Infomaster een aaneengesloten verschuivend blok is, moest een apart programma worden geschreven.

Daartoe wordt Infomaster geladen;

regel 714 en verder verwijderen

toevoegen regel 1000 - 1100 volgens programma onder no.?????

opgenomen in het drukwerkarchief

laad nu het bestand door 2 maal \*LOAD"filenaam" in casu voor de 'record'- en 'ascii'-file

- run infomaster en geef bij de vraag KEUS? >(escape)

- type in G.1000;(return) en wacht op de tekst

verplaats de ASCII file zoals wordt opgegeven door het programma mbv 'reloc' of 'copy' uit de Josbox of een equivalent daarvan

- laad het oorspronkelijke Infomaster-programma.

## 2 Het formatteren van de output

Formatteren of vormgeven van de output van Infomaster, is voor allerlei toepassingen interessant. Bijvoorbeeld:

- items verdelen over meerdere regels

- bij verdeling over meerdere regels de 2e regel laten inspringen

- printen van etiketten ed

### Items verdelen over meerdere regels

Wil men bv item 1 t/m 4 op de eerste regel en item 5 t/m 8 op de tweede regel of alle items op een afzonderlijke regel, dan is dat mogelijk door de navolgende regels aan het Infomaster-programma toe te voegen.

regel 300 : toevoegen ;K=8

regel 303 : 5 wordt 6

regel 309 : L.S wordt VALS

regel 567 : label x weghalen

Toevoegen:

regel 280 IFI=5P."LF"

regel 565:IFP?J&16;P.';U?1=U?1+I;C=0

Bij keuze 4 uit het printmenu verschijnt nu tav ieder item ook de vraag 'LF'. Bij N gaat alles zijn oude gang. Bij J verschijnt het desbetreffende item op een nieuwe regel en de daarna volgende lopen daar gewoon achteraan.

### Inspringen bij verdeling over meerdere regels

Voor de overzichtelijkheid van tekst kan het aantrekkelijk zijn op de tweede regel van een record in te springen, ofwel ergens in een regel een 'open-vlak' te creëren. Dit kan bereikt worden door bij de eerste aanmaak van de database een tweetal items toe te voegen die als 'naam' een 'spatie' krijgen.

Bij het opmaken van de output, worden dan mbv 'keuze 6' uit het printmenu, deze lege items op de juiste plaats gemanoeuvreerd en vervolgens bij 'keuze 4' uit het printmenu voorzien van de gewenste regellengte, waarmee de lengte van het inspringen wordt bepaald. Natuurlijk heeft deze werkwijze het nadeel dat bij het

of INFOBOL hebben wellicht de behoefte het databestand over te brengen naar Infomaster. Het is mogelijk dat de database vullen van de database, deze 'lege' items verschijnen, maar met het speciale commando kan daar gemakkelijk 'overheen' gesprongen worden.

#### Het printen van etiketten

Als de voorgaande werkwijzen is gevolgd, is het ook mogelijk om daadwerkelijk etiketten te printen. Er kunnen immers een aantal 'lege regels' worden gegenereerd op het juiste beginpunt voor het volgende record.

Men voegt dan mbv 'keuze 6' van het printmenu een of meerdere lege items achter elkaar, en geeft met 'keuze 4' van het printmenu bij 'LF' J

Voor degene die reeds Infomaster gebruikt is er dan nu het volgende probleem, om alsnog (lege) items aan bestaande bestanden toe te voegen. Geduld, er is al een hulpprogramma door Arie gemaakt, maar er zit nog een hardnekkige 'bug' in. Zodra deze eruit is wordt U geïnformeerd.

#### Het optellen van itemwaarden

Voor diegenen die infomaster toepassen op een magazijnbestand, kan het handig zijn de waarden die een bepaald item in alle records heeft, te kunnen optellen.

- verwijder regel 714 en volgende  
- voeg toe regel 714 t/m 754 volgens onderstaande listing

```
714GOS.690;IN."WELK ITEMNUMMER OPTELLEN"%S;IFL.SK1;G.A  
724J=VALS;IFJ>A;G.714  
734X=0;I=RR0;DOR=T+D?J;GOS.g  
744X=X+D;I=T+B;U.T>=RR1  
754P."TOTAAL "$NNJ" = "X";G.a
```

- RUN infomaster en kies "merge"(7)

#### Ofwel:

Verwijder de keuze 8 uit infomaster, kort een paar woorden wat in, geef aan de "optelroutine" een hoger regelnummer dan kan deze routine nog net erbij gevoegd worden (met bv. keuze 8 of 9)

#### 4. Enkele wetenswaardigheden:

##### a. Verkort Infomasterprogramma

Voor diegenen die een bestaand databestand willen "gebruiken", of door anderen laten gebruiken (bv. alleen sorteren, selecteren, opzoeken en afdrukken) dus zonder het bestaande te veranderen, kunnen evt. gebruik maken van een verkorte versie van Infomaster, die precies past in 4K, dus in een EPROM.

Deze verkorte versie heeft voornamelijk helaas het nadeel dat de "merge"-faciliteit ontbreekt.

Indien hiervoor belangstelling bestaat zal ik deze versie in het bandjesarchief laten opnemen. (B.T.)

b. Ten behoeve van de echte "knutselaar" is in het drukwerkarchief onder no. 116 beschikbaar een overzicht van alle in het Infomasterprogramma gebruikte variabelen

(adres, naam, omschrijving, voorbeeld van inhoud ), dit als service van Arie voor diegenen die zelf willen sleutelen aan het programma.

Met vriendelijke groet Bruno Tossaint (red)

\*\*\*\*\*  
MDCR GEINTERESSEERDEN  
\*\*\*\*\*

Geïnteresseerden in een MDCh kunnen zich tot uiterlijk 15 april opgeven bij: M.GRAUS 04498-52613. Meer info is ook bij hem te verkrijgen.

\*\*\*\*\*  
INHOUD BRONSGROEN BIKELTJE jaargang 2 nummer 3  
\*\*\*\*\*

- 2 Knipperende cursor met autorepeat en pieper : Hubert Vogten
- 3 Ledenlijst Acorn Computerclub Limburg
- 7 Bericht van het drukwerkarchief : Bruno Tossaint
- 8 Beginnershoekje : Evert Sanders
- 11 Lprom programmer soft : J. Salden
- 12 Bericht voor beginners en geïnteresseerden : Bruno Tossaint
- 13 Schijven-archief : Wim Ernst
- 13 Head - Data - kestore : Wim Ernst
- 15 Grapje van Hoenen : Henk Hoenen
- 16 Infomaster : Bruno Tossaint
- 19 MDCh geïnteresseerden : M. Graus
- 19 Inhoud Bronsgroen Bikelkje 11-3

Copy voor het regio-blad  
Demonstratie op een regio-avond  
het is en blijft welkom!!  
kunen we ook een keer op  
U rekenen?