

sinclair *gids*

NUMMER 3 / Februari 1987

PRIJS fl 6,50 / 130 BF

ONAFHANKELIJK BLAD VOOR SINCLAIR GEBRUIKERS

SCHAATSPROGRAMMA VOOR ZX SPECTRUM

TEST: MUSIC MACHINE

REKENEN MET ZX81

**WIN KLEURENMONITOR
IN AMX-MOUSE
PROGRAMMEERWEDSTRIJD**

**COMPACTER VOOR QL (slot)
LISTINGS VOOR SPECTRUM & QL**

**UITSLAG
1ste ronde ONE-LINER COMPETITIE**

Maandblad (8 x per jaar) verschijnt niet in januari, april, juli, oktober



CST

THOR

Eindelijk ook in Nederland leverbaar.
De capaciteiten van de Sinclair QL
komen nu pas echt goed
tot hun recht. De CST THOR is leverbaar
in 3 uitvoeringen, inclusief
X-change-software.

**KOMIN IS
DE OFFICIELE
IMPORTEUR
VAN**

CST THOR EN SANDY

SANDY

**VOOR HOBBYIST
EN PROFESSIONAL
SANDY Q-XT640**

VERKRIJGBAAR ALS BOUWKIT OF
ALS KOMPLEET
WERKEND SYSTEEM
MET 1 OF
2 DISKDRIVES

Model	specificatie	prijs	best.nr.			
1FW	20 MB Harddisk + 3 1/2 inch 720k diskdrive	f 6200,00 *	Q3500	Sandy Q-XT640	Compleet met 2 drives	f 2995,00 Q4020
1F	1 x 3 1/2 inch diskdrive	f 2660,00 *	Q3450	Sandy super- Q-board	512 Kram diskinterface	f 1198,00 Q750
2F	2 x 3 1/2 inch diskdrive	f 3150,00 *	Q3400	Sandy super- Q-board	0 Kram diskinterface	f 599,00 Q755
				Sandy interne	Geheugenuit- br. 512K kit	f 395,00 Q760
Het systeem wordt o.a. geleverd met:				Sandy Upgrade	Tot 512 Kram voor Q755	f 599,00 Q765
• 640 Kb RAM en 128 KB vrije EPROM- ruimte • systeemklok met batterij-backup • Qdos Ope- rationsysteem en Superbasic in ROM • Centronics parallel printerpoort • 2 seriële rs232 poorten • 2 joystick poorten • 1 muispoort • ICE software • 60 watt voe- ding ingebouwd enz. enz.				Sandy 512 Kram	Kaart	f 599,00 Q770
				Gigasoft	Muissysteem	f 395,00 Q780
				Gigachrome Software		f 175,00 Q110
				Front Page QL	deskkoppublisher (WYSIWYG)	f 135,00 Q047
				Better Basic	Verbeterd structuur en syntacs	f 96,00 Q048
				The Editor	10-20 sneller dan Quil	f 120,00 Q050
				Eye Q	Definitief het beste tekenprogramma	f 145,00 Q045
				Mediamanager	Cartridge- en diskdoctor	f 195,00 Q046
alle prijzen zijn inclusief BTW						

Nieuwe SINCLAIR QL Hardware

Model	specificatie	prijs	best.nr.
Sandy QXT640	Kit	f 1195,00	Q4000
Sandy Q-XT640	Compleet met 1 drive	f 2535,00	Q4010

KOMIN COMPUTERSYSTEMEN

Postbus 1805, 5602 CA Eindhoven, De Greefstraat 15A, 5622 GJ Eindhoven
Telefoon 040 - 456660*, Telex 59032 Notel NL
Bank: Rabo 15.90.05.701, A.B.N. 52.82.75.615, Giro: 46.43.401.

**NEDERLANDS GROOTSTE
sinclair SPECIALIST**

VRAAG ONZE GRATIS CATALOGUS

ONZE SHOWROOM IS GEOPEND MAANDAG T/M VRIJDAG VAN 9.00 TOT 17.00 UUR

Verzending
onder
rembours of
bij
vooruitbetaling.
Wijzigingen
voorbehouden.

Informatie
en bestellen
kan ook
telefonisch
tussen

10.00 en 16.00 uur
telefoon 040 - 456660

COLOFOON

De SINCLAIR GIDS is een uitgave van:
Terminal Software Publicaties
Postbus 111, 5110AC Baarle Nassau.
De SINCLAIR GIDS verschijnt 8x per jaar.

Hoofdredactie: P. Pauwels
Vaste medewerkers:

J.C.Deering E.Ragas
M.Feenstra A.Versluis
A.Pistorius A.v.d.Wijdeven
A.Plomp E.Zwart

ABONNEMENTEN:

NEDERLAND: fl. 44,00 per jaar (8 nummers)
Abonnementsgeld te voldoen door fl 44,00
over te maken op GIRO: 5109074 tnv. De
SINCLAIR GIDS, Baarle Nassau.

BELGIE: 860 BF per jaar (8 nummers).
Abonnementsgeld te voldoen door 860 BF
over te schrijven op POSTrekening :
000-1592677-34 tnv. De Sinclair Gids,
Baarle Nassau.

ABONNEMENTEN worden tot wederopzegging
aangegaan. Opzegging kan uitsluitend
schriftelijk geschieden en wel tot 1
maand voordat het huidige abonnement af-
loopt. Nadien vindt automatisch verlen-
ging plaats voor 1 jaar. Bij automatisch
verlengen dient U te betalen door middel
van de toegezonden acceptgiro.
ADRESWIJZIGINGEN 3 weken van te voren
opgeven met vermelding van oude en
nieuwe adres.

LOSSE NUMMERS: fl. 6,50 / 130 BF

De SINCLAIR GIDS is verkrijgbaar bij
boek- en tijdschriftenhandelaren, kios-
ken, grootwinkelbedrijven en bij compu-
terwinkels in Nederland en België.

INGEZONDEN PROGRAMMA'S

De inzender van programma's voor publi-
catie verklaart dat de programma's eigen
werk zijn en vrij van rechten. Hij/Zij
vrijwaart de SINCLAIR GIDS en uitgeverij
Terminal Software Publicaties voor enige
aanspraak van derden op het copyright
van desbetreffende programma's. Door in-
zending van een programma verklaart de
inzender dat hij op de hoogte is van
deze clausule.

COPYRIGHT: Het is NIET toegestaan arti-
kelen, of delen ervan over te nemen, zon-
der voorafgaande SCHRIFTELIJKE toestem-
ming van de uitgever. Programma's mogen
uitsluitend voor eigen gebruik worden
overgenomen.

Distributie Nederland:
Betapress, Burg.Krollaan 14, Gilze
Distributie België:
Persagentschap Vervoer en Distributie
Klein Eilandstraat 1, 1070 Brussel.

INHOUD

NEW : DOS voor de Spectrum	2
Exit Opus Discovery	3
INPUT : vragen en brieven	4
AMX-MUIS programmeer-wedstrijd	6
SERIES	
ZX81 : een nieuw begin	7
PEEK & POKE deel 3 en slot	12
TASWORD 2 : save/load graphics	44
TEST : de Music Machine van RAM	13
LLIST	
1. Spectrum	
Schaatsen met de Spectrum	15
Luxe LLister	21
Hex naar decimaal en vice versa ...	26
2. QL	
File compacter deel 2 en slot	27
ONE LINER competitie	
Uitslag van de 1ste ronde	37
VERENIGINGEN EN CLUBS	39
PROFIEL : Data-Skip uit Gouda	40
CASSETTE- en BOEKENSERVICE	42
VERIFY : The great Escape	43
IN/OUT : gezocht en aangeboden	46

WIN EEN
KLEURENMONITOR
ZIE BLADZYDE 6

EDITORIAAL

Het eerste nummer van dit nieuwe jaar. Net zoals iedereen hebben wij goede voornemens gemaakt. Eén ervan was, de Sinclair gids typografisch wat beter te verzorgen. Vandaar dat de tekst voortaan met een margrietwiel-printer wordt afgedrukt. Dat leest iets prettiger, dachten we.

De foto op de frontpagina heeft te maken met een programma op de middenbladzijden dat u toelaat om bij de komende schaatskampioenschappen alles netjes te volgen en te systematiseren. Dat had de heer Peters uit Almelo precies goed getimed.

Er is weinig ZX-nieuws te melden, deze keer. De +2 blijkt in Engeland prima te lopen, maar de Benelux loopt zo te zien wat achter. Misschien een reden om deze nieuwe telg iets meer te promoten?

In dit nummer wordt Data-Skip uit Gouda onder de loupe genomen. Lees hoe een klein bedrijf toch groot kan zijn.

De lijst met de gebruikersgroepen begint aardig te groeien. Sinclair blijkt zo te zien nog een hele schare aanhangers te hebben. En terecht, vinden wij.

Goed nieuws voor de ZX81-mensen: Han van Abbe zal er voor zorgen dat ook hun programmeer-honger wordt gestild. Hij start met een serie over "Rekenen met de ZX81" waarbij hij aan de hand van praktische voorbeelden de weg toont doorheen het inwendige van deze niet kapot te krijgen machine.

Wij waren zelf zo enthousiast over de mogelijkheden van de AMX-muis, dat we samen met Data-Skip besloten hebben om een programmeerwedstrijd te organiseren. Verderop in dit blad lees je alles over de prijzen die eraan verbonden zijn. Begin alvast te watertanden: wie zou er niet graag een kleurenmonitor hebben?

De kleine In/Outjes blijken ook aan te slaan. Zo te zien is de Sinclair-markt volop in beweging. Wie zijn/haar tekst niet terugvindt, herinneren we aan de spelregels: geen commerciële advertenties en geen gecopieerd spul.

Veel leesplezier!

SPECCY
DOS 4.0

Er is kortgeleden een disk operating system (DOS) ontwikkeld dat qua mogelijkheden niet moet onderdoen voor dat van Opus of BetaDisk.

Het DOS kan vier diskdrives besturen, waarvan één als "default" gekozen kan worden. Je kan om het even welk soort van drive gebruiken, omdat het systeem op ongeveer alle bekende formaten kan formatteren: van 35 tot 80 tracks, enkelzijdig of dubbelzijdig, enkele of dubbele dichtheid.

Speccy 4.0, zoals het systeem heet, is ook samen met Interface I te gebruiken, mits een paar kleine ingreepjes gedaan worden. Door middel van een simpele POKE kun je dan heen en weer schakelen tussen het DOS en de Interface I-ROM. Dat is enorm handig om bijvoorbeeld alle microdrive-programma's op schijf te zetten, zonder de langdurige omweg via cassette te hoeven gebruiken.

Om toe te laten, een indruk te vormen van de mogelijkheden van het systeem, geven we hieronder een overzicht van de commando's waar het systeem aan gehoorzaamt.

LOAD* heeft alle mogelijkheden van het gewone LOAD-commando.

SAVE* heeft eveneens alle mogelijkheden van de gewone SAVE. Daarbij kun je, mits toevoeging van de letter "d", een bepaalde belangrijke file twee keer op de schijf zetten, met 1 commando.

RUN* is een nieuwigheid. Dit commando laadt een programma of een code-blok van disk, en start het ook op, eventueel op een adres dat in het commando wordt megegeven, in de vorm: RUN*"naam"CODE,adres.

MERGE* werkt zoals het gewone MERGE-commando.

LIST* geeft een overzicht van wat er op een schijf staat. LIST*s geeft de vrije ruimte op de schijf aan. LIST*i geeft een lijst van de "onzichtbaar" gemaakte files op de schijf.

LLIST* doet hetzelfde als LIST*, maar nu wordt de informatie naar ZX-printer of de Alphacom.

ERASE* wist een bepaalde file van de schijf.

COPY* laat toe om back ups te maken, ofwel naar dezelfde drive, ofwel naar een andere.

LEN* geeft de vrije geheugenruimte op.

FORMAT* formatteert, zonder parameters, een schijf op 40 tracks, SSDD. Met toevoeging van de letters s of d voor enkel- of dubbelzijdig en enkele of dubbele schrijfdichtheid, plus een getal voor het aantal sporen (35,40,45,80) kan je zelf bepalen hoe de schijf wordt geformatteerd.

FN* wijzigt de naam van een bepaalde file op de schijf.

DEF FN* kan de naam van de schijf wijzigen.

AT* beveiligt de schijf tegen overschrijven.

ATTR* kan een bepaalde file beveiligen tegen overschrijven, en maakt ze tegelijk onzichtbaar in de catalogus.

USR* "naam" laat toe om bij het opstarten van het systeem, een bepaalde file automatisch in te laden en te runnen.

MOVE* maakt een back up van de hele schijf, naar dezelfde drive of naar een andere.

ASN* maakt één van de vier aangesloten drives tot "default" drive.

STEP* laat toe om de snelheid waarmee de kop zich tussen twee sporen verplaatst, aan te passen aan de diskdrive die je gebruikt.

Heel wat mogelijkheden, vinden wij. Het is jammer dat het ding nogal laattijdig op de markt verschijnt, en wat aan de prijzige kant is. Zeker nu er de concurrentie van Opus en de Disciple is. Maar het is wel superieur aan beide, wat de mogelijkheden betreft.

Leverancier : Micro-Connection,
St.-Katelijnevest 16-18, 2000 Antwerpen
Tel. 03/231 01 03
Prijs : 9990 fr/ca. f.549.-

EXIT OPUS DISCOVERY

OPUS heeft bekend gemaakt dat de Opus Discovery NIET meer geproduceerd zal worden. De recente prijsdaling (fl.495) was de laatste stuip trekking van een zeer populair disc-drivesysteem voor de ZX Spectrum.

Deze bekendmaking van OPUS heeft direct de verkoopprijs doen stijgen tot fl.595. Reden hiervan is dat de firma OPUS zelf geen voorraad meer heeft (en krijgt) en dat kleine partijen hier en daar in Europa moeten worden opgekocht. Dit uiter-aard tegen een hogere prijs.

Oorzaak van deze produktiestop is niet geheel duidelijk. OPUS zelf zegt dat na annulering van een order van 6000 stuks, bestemd voor Italië, men met een enorm overschot zat en men toen heeft besloten om de Discoveries uit te verkopen.

Andere medewerkers van OPUS echter melden, dat de markt voor de Discovery verzadigd raakte. ("een ieder die geïnteresseerd is heeft 'm inmiddels al!!"). Ook is bekend dat OPUS de Discovery nooit erg serieus heeft genomen: De Discovery voor de Spectrum deden ze, naast hun bestaande activiteiten, als "grapje" erbij.

Achteraf moge duidelijk zijn dat het onmogelijk is om voor fl.495 een compleet disk-systeem, inclusief drive, printerpoort en voeding te produceren. (Ter vergelijking: een diskdrive met interface voor de QL kost nog altijd fl.895.)

OPUS-bezitters hoeven, hoe dan ook, niet ongerust te worden. Er zijn inmiddels in Nederland en België, honderden gebruikers. En deze kunnen altijd een beroep doen op de leveranciers voor service en garantie. Ook de Discovery Users Club zal een ieder graag met raad en daad bijstaan. Toch is het jammer dat een diskdrivesysteem voor de ZX Spectrum (in 't begin fl.995) binnenkort niet meer leverbaar zal zijn.

(A.V.)

ZX SPECTRUM (+) SOFTWARE BOEK

Auteurs: P. Pauwels, M. de Chaffoy

Dit 192 pagina's dikke program-maboek voor de Spectrum 48K bevat vele nuttige programma's, zoals: A4 COPY, COPY RS232, VAL-maker, SCREENCOMPRESSIE en -EXPANSIE, FIND en REPLACE enz.

Verschijnt medio februari 1987.

ISBN 90-6883-027-9 fl.34,90

Verkrijgbaar in de goede boekhandel, Sinclair Speciaalzaken en Computercollectief.

TERMINAL SOFTWARE PUBLICATIES

In deze kolommen publiceren wij vragen van lezers, die algemeen genoeg zijn om ook andere Sinclair-fans te kunnen interesseren. Een reden daarvoor is, dat we willen proberen om via dit tijdschrift een dialoog op gang te brengen tussen de Sinclair-gebruikers. Een andere reden is, dat wij niet op elk gebied deskundig kunnen zijn en daarom bepaalde vragen niet zelf kunnen beantwoorden. Wij beschikken nu eenmaal niet over elk stuk software en hardware dat voor de Spectrum bedoeld is of er op aan te sluiten zou kunnen zijn. In deze rubriek worden ook brieven gepubliceerd, die voor andere lezers interessant zijn.

1. T. Witham wil weten of de software voor Viditel (geschikt voor de Spectrum 48K) ook op de 128K te gebruiken is en of de RS232-poort dan ook door die software in werking wordt gesteld.

2. Bij gebruik van het NOS Basiccode-vertaalprogramma 2a voor de Spectrum 48K in samenwerking met de printer Fasttext80 van Smith Corona, en de printer software voor de parallel-interface van MSTRonics blijkt dat na het inladen van de Basiccode, de printer-software spoorloos verdwenen is. Diverse load-volgordes werden geprobeerd, met steeds hetzelfde resultaat.

3. R. Verhoeff uit Den Haag wil de combinatie Masterfile + MFPrint + de Centronics GLP seriële printer + Interface I aan de praat krijgen. De handleiding bij de MFPrint vindt hij maar niks (kunnen we inkomen!). Wie kan stap voor stap uitleggen hoe MF Print aangepast kan worden?

4. R. Duijkers uit Epe wil een Marathon joystick aansluiten op een ZX Lprint Centronics RS232 interface. Kan dit, en zo ja: hoe dan? Hij bouwt ook een programmeerbare parallel in/out interface, uit het boek "ZX Spectrum Hardware Erweiterungen" van Schuessler (uitg. Data Becker). Weet iemand of die interface ook voor de Marathon joystick te programmeren is, en hoe?

5. De Spectrum 128 K van L. Casula uit Deventer heeft frekvent reset-neigingen, op de meest onvoorspelbare tijdstippen, maar wel altijd wanneer een programma loopt. Wie kent oorzaak en remedie?

6. VROUWEN EN COMPUTERS: GAAT DAT SAMEN?

Het traditionele denken, dat vrouwen niets van computers, programmatuur en randapparatuur weten en kunnen weten is al lang achterhaald. Hoewel ... Er is een groot struikelblok: vrouwen denken dat ze een heleboel WISKUNDE gehad moeten hebben om met deze apparatuur om te kunnen gaan. Men denkt dat ze alleen maar gegevens mag invoeren (datatypiste, heb je ooit al van een data-typist

gehoord?) en dan alleen als ze een "opleiding" gehad heeft van een uur. De BAAS kent alleen het wachtwoord en als hij ziek is mag ze er niet aankomen.

Voor zover bekend is het hersenweefsel van een vrouw op dezelfde manier opgebouwd als dat van een man, dus er is in wezen geen enkele belemmering voor haar om wat meer in de materie onder te duiken. Dat ze het niet doet, heeft verschillende oorzaken:

- 1- diep gezucht achter haar rug, maar wel zo dat ze het hoort;
- 2- een nog dieper gewortelde indoctrinatie, dat ze toch maar fouten maakt en zelfs een programma naar de bliksem helpt;
- 3- afpakneigingen van een in de buurt zijnde man (de goeden niet te na gesproken;
- 4- conservatisme (typmachines zijn voor vrouwen gemaakt);
- 5- geen interesse.

Aan de laatste twee punten is misschien iets te doen.

Ik loop wel eens door een grote middelbare school met een draagbare computer en dan krijg ik stevast de vraag of ik alweer met de naaimachine loop te zeulen. "Ik neem hem wel even over. Waar moet je naartoe?". De mannelijke conrector, die even vaak met het ding loopt te sjouwen, krijgt die vraag nooit ...

Als een vrouw zich met computers bezig houdt, heeft ze volgens de rest van de wereld minstens een intelligentie van 140 en te veel mannelijke hormonen. Ze is apart. Iets voor de kermis?

Tips voor vrouwen om dit patroon te doorbreken:

- 1- Schaf een computer aan (een Spectrum of zo).
- 2- Lees de gebruiksaanwijzing GOED door. Het staat niet goed als je alles moet vragen.
- 3- Programmeer iets leuks in BASIC. En doe het alleen (dus geen pottekijkers).
- 4- Sluit aan bij een gebruikersgroep.
- 5- Ga naar de bijeenkomsten van die gebruikersgroep. Mijn ervaring is dat je aan alle kanten al of niet gevraagde hulp krijgt.
- 6- Maak vrienden onder de computer-freaks. (Die komen trouwens vanzelf).
- 7- Maak je geen zorgen als het niet meteen lukt. Iedereen moet wennen en leren. (lees verder op blz. 42)



De specialist
in groot en klein,
hard en zacht.
Wij voeren zowat
alle merken, maar
vooral Sinclair
en aanverwanten.
Bel of schrijf ons
voor prijzen en info.

MICRO-CONNECTION

St. Katelijnevest 16/18
2000 Antwerpen
Tel. (03)231 01 03-231 15 40

In samenwerking met de firma Dataskip uit Gouda organiseert de SINCLAIR GIDS een programmeerwedstrijd rondom de AMX-Muis.

Waarom deze programmeerwedstrijd ?

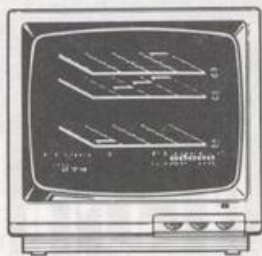
De AMX-muis is een nieuw medium voor de Spectrum, dat geheel nieuwe mogelijkheden aan Uw Spectrum geeft. Omdat er in feite weinig informatie en programma's speciaal voor de AMX-Muis worden gepubliceerd is deze wedstrijd georganiseerd.

De programmeerwedstrijd loopt van 1 februari t/m 30 april 1987. De uitslag van deze wedstrijd wordt gepubliceerd in de Sinclair Gids no. 7.

WIN EEN KLEURENMONITOR

AMX *mouse*

PROGRAMMEERWEDSTRIJD



DE PRIJZEN

Voor deze wedstrijd zijn de volgende prijzen beschikbaar gesteld:

- | | |
|--------------------------|-----------------|
| 1e prijs: Kleurenmonitor | twv. fl. 700,00 |
| 2e prijs: Videoface | twv. fl. 275,00 |
| 3e prijs: VTX-5000 MODEM | twv. fl. 149,00 |
| 4e prijs: KONIX Joystick | twv. fl. 39,00 |
| 5e prijs: KONIX Joystick | twv. fl. 39,00 |

6e t/m 10e prijs:
ZX Spectrum Software Boek twv. fl. 34,90

1e prijs en 6e t/m 10e prijs beschikbaar gesteld door de SINCLAIR GIDS. 2e t/m 5e prijs beschikbaar gesteld door DATASKIP, Gouda.

REGLEMENT

1. De deelnemer accepteert door inzending van zijn deelnemend programma automatisch de wedstrijdvoorwaarden.
2. Het ingezonden programma dient eigen werk te zijn. De deelnemer vrijwaart de Sinclair Gids / Uitgeverij Terminal Software Publicaties van aanspraken van derden op de ingezonden programma's.
3. Medewerkers van DATASKIP en Uitgeverij Terminal Software Publicaties zijn van deelname uitgesloten.
4. De programma's moeten uiteraard werken met de AMX-muis en de ZX Spectrum.
5. De inzendingen moeten uiterlijk 30 april 1987 bij de redactie van de Sinclair Gids binnen zijn.
6. Alle inzendingen worden eigendom van de Sinclair Gids.
7. De inzending dient te geschieden op cassette of disk (OPUS). De uitleg als Tasword-file op cassette of geprint of getypt op A4.
8. De jury bestaat uit T.Versluis, P. Pauwels, E.Ragas en als extra neutraal lid Mevr. N.Ragas, die de gebruikersvriendelijkheid van de programma's zal toetsen.
De originaliteit en kwaliteit zullen van doorslaggevende betekenis zijn.
9. De uitslag wordt gepubliceerd in de Sinclair Gids nummer 7. Bovendien zullen in dat nummer de winnende programma's worden gepubliceerd. Over de uitslag is geen correspondentie mogelijk.



BELANGRIJK

Belangrijk om te weten is dat met ingang van nummer 4 een tweede programmeerwedstrijd gaat lopen. Deze wedstrijd zal betrekking hebben op toepassing van de Spectrum in het onderwijs als leerhulpmiddel. Deelnemers aan de AMX-muis wedstrijd kunnen dus met hun inzending tegelijkertijd deelnemen aan de nieuwe wedstrijd, uiteraard moet het dan een educatief programma zijn.

SNEL EN PRECIES REKENEN MET DE ZX 81

Snel en precies rekenen met de ZX81. Onder deze titel zal ik een aantal artikelen publiceren met programma's die ik sinds 1982 heb ontwikkeld. Daarbij wordt meestal het toepassen van de

FLOATING POINT CALCULATOR

vermeden, omdat die traag en onnauwkeurig is. Maar niet altijd, soms kan er toch een nuttig gebruik van worden gemaakt. Hetzelfde geldt voor de

FUNKTIES en de OPERATOR **

van de ZX 81.

Als eerste voorbeeld een recente ervaring. In het PROBLEEM VAN DE MAAND NOVEMBER HCC NIEUWSBRIEF NR.87 komen 5-de machten voor. En aangezien het een eenvoudig probleem betreft kan het met een BASIC programma worden opgelost. Natuurlijk ben je dan geneigd om de OPERATOR ** te gebruiken. Maar die werkt onnauwkeurig zoals het programma van figuur 1 aantoont. Zie ook de printout.

Fig. 1 ONNAUWKEURIGHEID VAN **

```
2 PRINT 2;" ";2*2*2*2*2-32,2
**5-32
3 PRINT 3;" ";3*3*3*3*3-243,
3**5-243
4 PRINT 4;" ";4*4*4*4*4-1024
,4**5-1024
5 PRINT 5;" ";5*5*5*5*5-3125
,5**5-3125
6 PRINT 6;" ";6*6*6*6*6-7776
,6**5-7776
7 PRINT 7;" ";7*7*7*7*7-16807
7,7**5-16807
8 PRINT 8;" ";8*8*8*8*8-32768
8,8**5-32768
9 PRINT 9;" ";9*9*9*9*9-59049
9,9**5-59049
```

RESULTAAT:

2	0	0
3	0	5.9604645E-7
4	0	0
5	0	1.9073486E-6
6	0	-1.9073486E-6
7	0	-7.6293945E-6
8	0	0
9	0	.0002746582

Er zijn twee oplossingen voor dit fenomeen.

1. De ALGEMENE manier voor het vergelijken van een misschien onnauwkeurig getal/uitkomst (X) met een exacte grootte (A):

gebruik niet IF X=A maar IF ABS (X-A)<1.

2. De in dit geval SPECIFIEKE manier: gebruik A*A*A*A*A ipv. A**5.

Het blijkt echter ook dat **5 veel meer tijd kost dan 5-maal vermenigvuldigen zoals timing van de beide programma's in fig. 2 laat zien. De tweede oplossing is dus duidelijk preferent.

De oplossing die ik van het HCC-probleem heb ingestuurd laat zien hoe 'slim' programmeren enorme tijdswinst kan opleveren. Terug naar ons probleem. Dit kan met een BASIC-programma worden opgelost.

De opgave was:

Vindt alle getallen van 5 cijfers abcde tussen 9.999 en 100.000, waarvoor geldt

$$abcde = a^5 + b^5 + c^5 + d^5 + e^5$$

Fig.2 TIMING **5 vs.

```
10 FOR X=1 TO 300
20 LET Y=X**5
30 NEXT X
40 STOP
50 FOR X=1 TO 300
60 LET Y=X*X*X*X*X
70 NEXT X
```

RESULTAAT:

Tijd met **	35 Sec
Tijd met *	5 Sec

De 'rechttoe rechtaan'-oplossing van figuur 3 zet eenvoudigweg de opgave om in ZX81-taal. Daarbij wordt rekening gehouden met wat ik hiervoor heb gezegd over de onnauwkeurigheid van de operator **, zie regel 40. Na 16 uur en 48 minuten in FAST-mode (!) worden de drie goede antwoorden geprint. Dit programma kost zoveel tijd, omdat alle 90.000

Fig. 3 HCC PROBLEEM B86 ***5 RECHTTOE RECHTAAN-PROGR.

```
10 FOR N=1E4 TO 99999
20 LET N$=STR$ N
30 LET S=VAL N$(1)**5+VAL N$(2)**5+VAL N$(3)**5+VAL N$(4)**5+VAL N$(5)**5
40 IF ABS (S-N)<1 THEN PRINT N
50 NEXT N
```


5-cijferige getallen abcde moeten worden getest. Bovendien zijn de trage functie VAL en de eveneens trage operator ** gebruikt.

Daarom in fig. 4 de beloofde 'slimme' oplossing met 'omgekeerde bewijslast': vorm alle wezenlijk verschillende combinaties van 5 cijfers 0...9 en onderzoek of de som V van de 5-de machten van die cijfers gelijk is aan de som X van de 5-de machten van de cijfers van V. Dan behoeven slechts 2002 combinaties te worden getest. Bovendien wordt -zoals reeds besproken- veel tijd bespaard door 5-maal te vermenigvuldigen i.p.v. ** te gebruiken. Na 534 seconden blijken er drie goede oplossingen te zijn:

93084

92727

54748

Fig. 4 HCC PROBLEEM B86 ***5"
SNELLE OPLOSSING

```

10 FOR A=0 TO 9
20 FOR B=A TO 9
30 FOR C=B TO 9
40 FOR D=C TO 9
50 FOR E=D TO 9
60 LET V=A*A*A*A*A+B*B*B*B*B+C
  *C*C*C*C+D*D*D*D+E*E*E*E
70 IF V<1E4 OR V>99999 THEN GO
  TO 150
80 LET V$=STR$ V
90 LET X=0
100 FOR N=1 TO 5
110 LET W=VAL V$(N)
120 LET X=X+W*W*W*W*W
130 NEXT N
140 IF X=V THEN PRINT V
150 NEXT E
160 NEXT D
170 NEXT C
180 NEXT B
190 NEXT A

```

Het kan nog wat sneller namelijk in 405 seconden door gebruik te maken van een tabel voor de 5-de machten van 2 t/m 9. Dan hoeft er minder te worden gerekend en dat betekent meestal tijdswinst. Die tabel wordt gevormd in de regels 10 t/m 50 van fig. 5.

Daar V(0) bij de ZX81 niet bestaat moeten alle grootheden 1 hoger worden genomen: A=1 TO 10 i.p.v. 0 TO 9., enz. De regels 30 en 160 zorgen ervoor dat alles toch op z'n pootjes terechtkomt.

Fig 5. HCC PROBLEEM B86 ***5"
OPLOSSING MET TABEL

```

10 DIM V(10)
20 FOR N=1 TO 10
30 LET W=N-1
40 LET V(N)=W*W*W*W*W
50 NEXT N
60 FOR A=1 TO 10
70 FOR B=A TO 10
80 FOR C=B TO 10
90 FOR D=C TO 10
100 FOR E=D TO 10
110 LET V=V(A)+V(B)+V(C)+V(D)+V
  (E)
120 IF V<1E4 OR V>99999 THEN GO
  TO 200
130 LET V$=STR$ V
140 LET X=0
150 FOR N=1 TO 5
160 LET W=VAL V$(N)+1
170 LET X=X+V(W)
180 NEXT N
190 IF X=V THEN PRINT V
200 NEXT E
210 NEXT D
220 NEXT C
230 NEXT B
240 NEXT A

```

PROBLEEM VAN DE MAAND JUNI 1986.

Als tweede onderwerp nog een cijferpuzzel. Dit keer behandel ik een oplossing in BASIC en een zeer veel snellere MACHINECODE routine. In beide programma's enkele trucs, die ook bij andere problemen bruikbaar zijn.

De opgave in HCC NIEUWSBRIEF NR.83 luidde:

Schrijf de getallen 2,3,...,8 en 9 als een breuk waarin de cijfers 1 t/m 9 elk 1-maal voorkomen. Als er meer mogelijkheden zijn, kies dan die met de kleinste getallen. De gepubliceerde oplossing staat in HCC NIEUWSBRIEF NR.87 van november 1986. De kern van die oplossing zit ook in mijn programma's:

Om een breuk te maken die alle cijfers van 1 t/m 9 bevat, terwijl de uitkomst (het resultaat R) een van de cijfers 2 t/m 9 is, moet de N(oemer) uit 4 cijfers bestaan en de T(eller) uit 5 cijfers. Voor iedere waarde van R is er een minimumwaarde van de noemer te vinden die de teller 5 cijfers maakt. Met die waarde beginnend berekenen we steeds $T=R*N$ en onderzoeken of alle cijfers 1 t/m 9 voorkomen in de T en N samen. Zoniet dan nemen we de eersthogere N die geen 0 en geen gelijke cijfers bevat. Vaststellen of alle cijfers 1 t/m 9 voorkomen doe ik anders dan in het geciteerde HCC-artikel, zoals bij de bespreking van de programma's zal blijken.

Mijn BASIC-oplossing staat in fig. 6. Allereerst valt op dat met die goeie ouwe ZX81 een veel korter programma mogelijk is dan het gepubliceerde.

De werking is als volgt:

- begin met R=2
- D\$ bevat de 9 bovengenoemde minimum-waarden van N die in regel 30 als string worden uitgelezen.
- X\$ bestaat uit 9-maal een X; voor de in N en T gebruikte cijfers, zie regels 90 en 150.
- in regel 50 wordt N steeds met 1 verhoogd.
- nullen en reeds gebruikte cijfers doen N verder verhoogd worden
- als een N is gevormd wordt T berekend, ook als string.
- vervolgens wordt die T van 5 cijfers behandeld als eerder de N en als hij die behandeling 5-maal overleeft -dus zonder naar regel 40 te zijn teruggestuurd- hebben we een oplossing gevonden, die wordt uitgeprint.
- de waarde van R wordt verhoogd en we beginnen opnieuw; dit gaat door totdat R=10.

Na bijna 82 minuten zijn de oplossingen bekend, zie de tabel van fig. 6.

Fig. 6 HCC PROBLEEM 686 "BREUK"
OPLOSSING MET BASIC PROG

```

10 LET R=2
20 LET D$="5123341225132134167
2143212531234"
30 LET N$=D$(4*R-7 TO 4*R-4)
40 LET X$="XXXXXXXXX"
50 LET N$=STR$(VAL N$+1)
60 FOR M=1 TO 4
70 IF N$(M)="0" THEN GOTO 40
80 IF X$(VAL N$(M))=" " THEN G
OTO 40
90 LET X$(VAL N$(M))=""
100 NEXT M
110 LET T$=STR$(R*VAL N$)
120 FOR M=1 TO 5
130 IF T$(M)="0" THEN GOTO 40
140 IF X$(VAL T$(M))=" " THEN G
OTO 40
150 LET X$(VAL T$(M))=""
160 NEXT M
170 PRINT T$; " / ";N$; " = ";R
180 LET R=R+1
190 IF R<10 THEN GOTO 30

```

RESULTAAT:

```

13458 / 6729 = 2
17469 / 5823 = 3
15768 / 3942 = 4
13485 / 2697 = 5
17658 / 2943 = 6
16758 / 2394 = 7
25496 / 3187 = 8
57429 / 6381 = 9

```

Omdat ik dat toch wel veel tijd vond voor dit probleem heb ik ook een MACHINECODE-programma gemaakt. Dat geeft me de gelegenheid enkele fundamentele routines te behandelen, die vaker -eventueel uitgebreid of gewijzigd- zullen voorkomen. Vooraf het resultaat: het programma van fig. 7 doet het werk in minder dan 12 seconden !! Ruim 400-maal zo snel en even precies als in BASIC. De trage en meestal niet preciese FLOATING POINT CALCULATOR is bewust vermeden. (Figuur 7 staat op blz. 10 e.v.)

De routines werken als volgt:

- DATA is het equivalent van D\$ uit regel 20 van fig. 6; de plaats in RAM is welbewust gekozen met L=01 t/m 10 HEX, om daar de waarden van B(=R-1) uit af te leiden.
- de START is op adres 5011 met RAND USR 20497 en na initiatie volgt het SETten van B, voor de eerste ronde is de waarde 08; de eerste MINIMUM VALVE die daar bij hoort is 1234.

- de voorbereiding voor het vermenigvuldigen gebeurt door de 4 digits (cijfers) van de noemer N uit het (HL)-gebied over te brengen naar 't teller-gebied T-STORE met de deelroutine TRFR N TO T om vervolgens de 1-ste (=HI) DIG van T 0 te maken.

- MULTIPLY, het uitvoeren van R*N=T, gebeurt door N=DEB-maal decimaal (DCM) met carry op te tellen (ADC) bij wat zich in T-STORE bevindt; de instructie DAA zorgt er voor dat dit inderdaad DCM gebeurt.

- daarna worden de 5 + 4 digits van T en N getest met behulp van de subroutine DIG-TO-BITS die digits in bits omzet en als die 9 digits de BITS-STORE ED= 0000 0001 llll hebben gemaakt zijn de bij de waarde van R behorende waarden van T en N gevonden, die aan de opgave voldoen.

- TRFR brengt zo'n goed antwoord naar de T N R-TABLE over, welke van rechts onder naar links regel voor regel wordt gevuld met van links naar rechts de teller, de noemer en het resultaat; voor de leesbaarheid van die tabel zijn de T- N- en R-STORES steeds door een NOP-byte gescheiden: adressen 4046 en 4049.

- zolang geen goed antwoord is gevonden wordt de waarde van N DCM met 1 verhoogd en wordt verder gezocht met NXT-N; nullen en reeds gebruikte cijfers worden hier niet 'geskippt' zoals in het BASIC programma, omdat de 8 antwoorden toch razendsnel bekend zijn; is wel een goed antwoord gevonden dan wordt met de naast lagere R via NXT-R het zoeken voortgezet.

Indien daaraan behoefte bestaat kan de inhoud van de T N R-TABLE net zo uitgeprint worden als in fig. 6 is gedaan. Hoe, dat zal ik in een van de volgende artikelen in deze serie behandelen.

Han van Abbe

Fig. 7 HCC PROBLEEM 0686 "BREUKEN"
OPLOSSING MET MACHINECODE

PTR(POINTER) TABLE T N R 4041/2

STORES/TABLES WITH DCM(DECIMAL)

DIG(ITS) HI...LO:

T(ELLER) STORE 4043/4/5

N(OEMER) STORE 4047/8

R(ESULTS) STORE 404A

DATA 5001/10

T N R-TABLE 50A0/DF

DATA 5001--5123 3412 2513 2134 R= 2,3,4,5
5009--1672 1432 1253 1234 6,7,8,9

LABELS	ADRES/MNEMONICS	Z80-INSTRUCTIE	COMMENTAAR
INITIATIE			
START	5011--CDE702	CALL 02E7	SET-FAST
	5014--21DF50	LD HL,50DF	HI-TABLE
	5017--224140	LD (4041),HL	PTR-TABLE
	501A--211050	LD HL,5010	HI-DATA
SET B + TRFR N TO T + CLEAR DIG			
NXT-R	501D--E5	PUSH HL	STACK DATA-AD
	501E--45	LD B,L	L/2=B-R-1=8.....1
	501F--CB28	SRA B	
NXT-N	5021--C5	PUSH BC	STACK B
	5022--114540	LD DE,4045	LD-DIG-T
	5025--7E	LD A,(HL)	
	5026--12	LD (DE),A	4./5.DIG-T=3./4.DIG-N
	5027--2D	DEC L	
	5028--1D	DEC E	
	5029--7E	LD A,(HL)	
	502A--12	LD (DE),A	2./3.DIG-T=1./2.DIG-N
	502B--1D	DEC E	
	502C--AF	XOR A	
	502D--12	LD (DE),A	CLEAR 1.DIG-T
MULTIPLY N WITH R BY B* DCM ADC			
NXT-ADC	502E--ED5B4440	LD DE,(4044)	DE=N
	5032--214540	LD HL,4045	LO-DIG-T
	5035--7A	LD A,D	
	5036--86	ADD A,(HL)	
	5037--27	DAA	
	5038--77	LD (HL),A	4./5. DIG
	5039--2D	DEC L	
	503A--7B	LD A,E	
	503B--8E	ADC A,(HL)	
	503C--27	DAA	
	503D--77	LD (HL),A	2./3. DIG
	503E--3002	JR NC,5042	
	5040--2D	DEC L	
	5041--34	INC (HL)	1.DIG
	5042--10EE	DJNZ 5032	NXT-ADC
TEST DIG T AND N + INCREMENT N			
NXT	5044--ED534740	LD (4047),DE	STORE N
	5048--2E42	LD L,42	BEGIN TEST AT 1.DIG-T
	504A--110000	LD DE,0000	CLEAR BITS-STORE
	504D--0E06	LD C,06	3 BYTES T+NOP+2 BYTES N
	504F--CD8550	CALL 5085	DIG-TO-BITS
	5052--0D	DEC C	

INC

5053--20FA	JR	NZ,504F
5055--A7	AND	A
5056--2101FF	LD	HL,FF01
5059--ED52	SBC	HL,DE
505B--C1	POP	BC
505C--280F	JR	Z,506D
505E--214840	LD	HL,4048
5061--E5	PUSH	HL
5062--3E01	LD	A,01
5064--86	ADD	A,(HL)
5065--27	DAA	
5066--77	LD	(HL),A
5067--2D	DEC	L
5068--38F8	JR	C,5062
506A--E1	POP	HL
506B--18B4	JR	5021

NXT
CARRY=0
LH=0000 0001 1111 1111
ED=BITS T&N
RETRIEVE B
TRFR
3./4.DIG-N

+1 DCM

INC 1./2.DIG IF CARRY

NXT-N

TRANSFER R N T TO T N R-TABLE

TRFR

506D--214A40	LD	HL,404A
5070--ED5B4140	LD	DE,(4041)
5074--04	INC	B
5075--70	LD	(HL),B
5076--010800	LD	BC,0008
5079--EDB8	LDDR	
507B--ED534140	LD	(4041),DE
507F--E1	POP	HL
5080--2D	DEC	L
5081--2D	DEC	L
5082--2099	JR	NZ,501D
5084--C9	RET	

R-STORE
PTR-TABLE
MAKE B=R
STORE B
TRFR(404A....4043)
STORE NXT-PTR
RETRIEVE DATA-AD
L-VALVE FOR
NEXT LOWER R
NXT R
TO BASIC IF L=00:R<2

CONVERT DIGITS INTO BITS

DIG-TO-BITS	5085--23	INC	HL
	5086--7E	LD	A,(HL)
	5087--CD8F50	CALL	508F
	508A--7E	LD	A,(HL)
	508B--1F	RRA)
	508C--1F	RRA)
	508D--1F	RRA)
	508E--1F	RRA)
RIGHT-DIG	508F--E60F	AND	OF
	5091--47	LD	B,A
	5092--AF	XOR	A
	5093--B8	CP	B
	5094--C8	RET	Z
	5095--37	SCF	
	5096--17	RLA	
	5097--10FD	DJNZ	5096
	5099--3001	JR	NC,509C
	509B--1C	INC	E
	509C--82	ADD	A,D
	509D--57	LD	D,A
	509E--C9	RET	

RIGHT-DIG

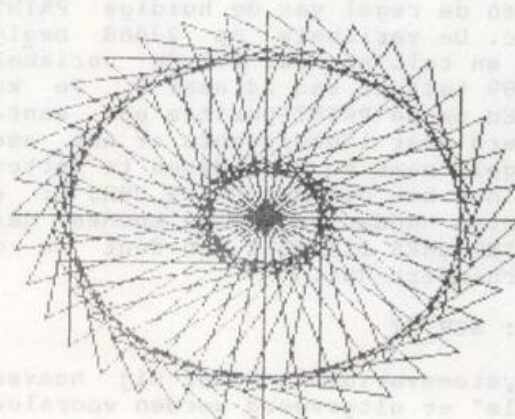
LEFT
DIGIT
TO
RIGHT

IF DIG=0
BEGIN WITH CARRY
A=2**(B-1)

SET BIT "9"
SET BITS "8....1"

T N R-TABLE

50A0--0134	5800	6729	0002
50A8--0174	6900	5823	0003
50B0--0157	6800	3942	0004
50B8--0134	8500	2697	0005
50C0--0176	5800	2943	0006
50C8--0167	5800	2394	0007
50D0--0254	9600	3187	0008
50D8--0574	2900	6381	0009



De ZX Spectrum Systeemvariabelen DEEL 3

23675 en 23676 : UDG

PEEKen van deze systeemvariabele, levert het adres op van de bovenste pixel-lijn van het eerste zelf gedefinieerde karakter, beter bekend als : USR "a". Er is overigens een gemakkelijker manier om hetzelfde resultaat te bereiken : het commando "PRINT USR "a"" geeft ook dat adres, met minder typwerk. Andersom gaat niet : POKE USR "a",0 is niet hetzelfde als POKE 23675,0 of POKE 23676,0.

Met deze systeemvariabele kunnen we twee kanten op. Als we de UDG's niet nodig hebben, doen we CLEAR 65535. Als we meer dan 21 UDG's nodig hebben, kunnen we er bij maken, hoewel we maar 1 groep tegelijk kunnen gebruiken. Elke set kost $21 \times 8 = 168$ bytes, dus moeten we RAMTOP zoveel maal 168 bytes verlagen als we sets UDG's willen bijmaken. Afhankelijk van welke groep we willen gebruiken, passen we het startadres aan, d.m.v. POKES in deze systeemvariabele.

23677 en 23678 : COORDS

Deze systeemvariabelen bevatten resp. de x-coördinaat en de y-coördinaat van de laatst geplote pixel. Het CLS-commando zet ze allebei op nul. Je kan elke gelidige x of y-waarde hierin POKEN. Daardoor wordt er op het scherm niets gewijzigd, maar het volgende DRAW-commando vertrekt wel van het gePOKEte punt. Dat is handig, als je van een punt wil vertrekken, zonder daar iets te plotten.

23688 en 23689 : SPOSN (screen position)

Deze bytes bevatten informatie over de kolom en de regel van de huidige PRINT-positie. De variabele op 23688 begint bij 33 en telt af naar 1; de variabele op 23689 telt af van 24 naar 3. Je kan ze POKEN om de PRINT-positie een aantal karakters naar links/rechts of een aantal regels naar boven/onderen te zetten. Naar links kan ook met PRINT CHR\$ 8; en naar boven hadden we moeten kunnen gaan met PRINT CHR\$ 11; maar een fout in de ROM verhindert dat.

23692 : SCR_CT

Deze systeemvariabele houdt bij hoeveel "scrolls" er uitgevoerd worden vooraleer

de "scroll ?"-vraag te stellen. Let op: 1 "scroll" is 1 regel. Als je dus POKet met 255, zullen er 255 regels gescrolld worden vooraleer de vraag komt. POKen met 0 geeft 1 regel meer.

23693 : ATTR_P

Hierin worden de globale scherm-attributen bewaard, die gebruikt worden na een CLS-commando of in een gewoon PRINT-commando. POKen heeft enkel zichtbaar effect na een PRINT of een CLS, maar het gaat soms sneller. Een voorbeeld : POKE 23693,56 is hetzelfde als PAPER 7 : INK 0: BRIGHT 0 : FLASH 0 maar gebruikt 11 bytes minder en werkt sneller (maar één commando in plaats van vier).

23694 : MASK_P

Deze byte wordt geraadpleegd indien de waarde 8 gebruikt wordt bij een attributes-statement. INK 8 zal bits 0, 1 en 2 setten. PAPER 8 set bits 3, 4 en 5. BRIGHT 8 set bit 6. FLASH 8 set bit 7 van deze systeemvariabele. Wanneer er iets te printen valt, komt de computer hier kijken (als er geen attributen worden gespecificeerd in het statement dat wordt uitgevoerd : zo ja, dan neemt hij natuurlijk die) om de attributen te bepalen. Dat kan leiden tot op het eerste gezicht rare effecten, omdat de computer de kleurwaarden voor PRINT bepaalt d.m.v. een AND-functie (de attributen op het scherm AND 8). Dat leidt gewoonlijk tot het overnemen van de kleuren op het scherm (bv. INK 4 = BIN 100; 8=BIN 111; 100 AND 111 geeft 100, d.w.z. terug INK 4). Wanneer we evenwel in deze systeem-variabele een ander bitpatroon POKEN, krijgen we andere kleuren. Stel, na het commando INK 8, POKen we 2 (d.w.z. BIN 010). Bij een PRINT-instructie zal de gebruikte kleur (bv. INK 5) geAND worden met 2. Het resultaat is niet 5, zoals verwacht, maar BIN 101 AND BIN 010 = 0 !

23730 en 23731 : RAMTOP

Op deze plek staat het hoogste adres waar Basic nog toegang heeft (de UDG's uitgezonderd). Je kan dat op elke willekeurige waarde zetten, hoewel er grenzen zijn : maximum 65535, minimum ca. 24000, tenzij je volledig in machinecode werkt. Onder dit adres komt immers niet enkel je Basic te staan, maar ook alle stacks, RETURN-adressen van GOSUBS, ook alle channel- en microdrive-informatie. Alles boven dit adres (niet 23730, maar het adres dat op deze 2 bytes staat !) is veilig voor NEW en CLEAR. Een meer voor de hand liggende manier om de RAMTOP op een zelf gekozen plaats te zetten, is het commando "CLEAR nn". (P.P.)

RAM

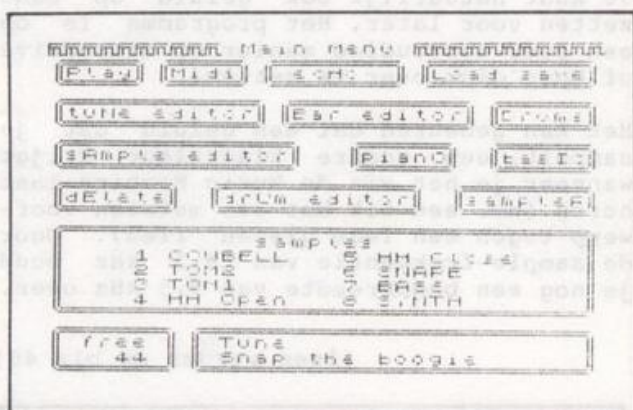
MUSIC

MACHINE

De RAM Music Machine ziet er net zo uit als de RAMPRINT. Behalve de achterkant. Daar zitten zes aansluitingen: koptelefoon, microfoon, audio-lijn naar een versterker, en drie MIDI-aansluitingen (in, doorvoer en uit). Bovenop zit nog een schuifregelaar voor het volume.

De aansluitstekker van de interface past op de I/O-bus van de Spectrum en op die van Interface I. Om ongevallen te voorkomen, is het doosje voorzien van een oog, zodat je de voedingsstekker er pas kan instoppen nadat de Music Machine werd aangesloten. Dan nog een versterker aansluiten (de audio uitgang is CINCH) en de meegeleverde microfoon en de pret kan beginnen.

Na het LOAD-commando heb ik de tijd om de handleiding door te bladeren. Er moet zo'n 38 K worden ingeladen. Na zowat 8 minuten zie ik het hoofdmenu. Behalve dat het op een aparte manier wordt geschreven en gewist (van links naar rechts en van rechts naar links) is het ook even wennen aan de bediening. Elke optie in dat menu bevat 1 hoofdletter. Door intoetsen van die letter kies je. Eenvoudig na een tijd, maar apart.

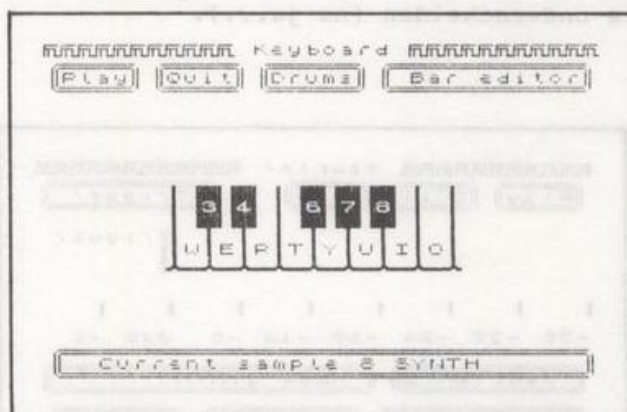


Op de cassette staan al acht geluiden, die met het programma mee worden ingeladen. Je kunt dus al direkt aan de slag.

Het instrument waarmee ik het meest vertrouwd ben, is de piano. Dus druk ik op de pian"O". Op het scherm verschijnt een octaaf van een pianoklavier. Op elke toets staat welke toets van de Spectrum ermee overeenkomt. Het is duidelijk voor de echte Spectrum toetsenborden bedoeld (de rubberen, de +, de 128 en de +2): de lay out van Saga 1 is anders, zodat halve tonen niet (vlot) te spelen zijn.

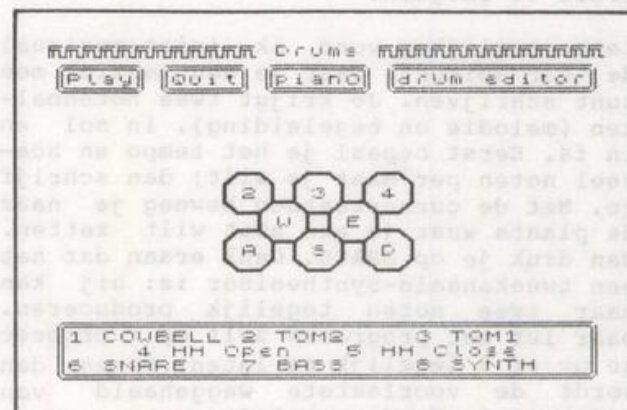
Even een deuntje spelen. Geen akkoorden, alleen enkelvoudige noten. Speelt wel vlot. De respons op de toetsen is goed.

Volgende stap is, het geluid "bekijken". Dat kan middels de "sample editor". Die werkt prettig. Je kan tot op de milliseconde nauwkeurig bewerken, inkorten, verschuiven, omkeren, stukjes aan elkaar plakken, kortom je kunt opgenomen geluid compleet anders doen klinken. Wanneer je een zin via de microfoon invoert (er is ruimte voor iets meer dan een seconde geluid) dan kun je er echt 1 letter, of 1 onder-deel van een klinker uitvissen, en die gebruiken. Mooi. Je krijgt ook continu informatie op het scherm over de grootte van het beschikbare geheugen, over de lengte van het sample dat je aan het bewerken bent, enzovoort.



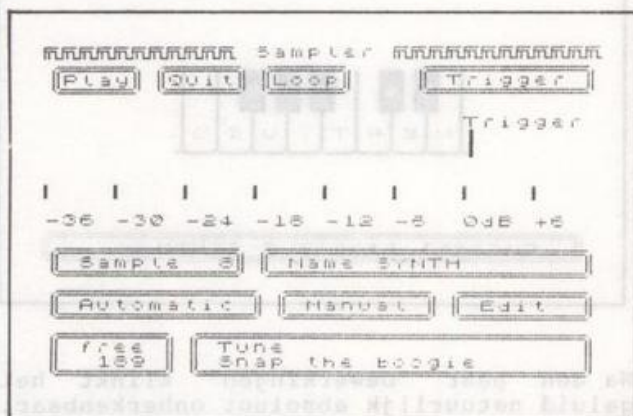
Na een paar "bewerkingen" klinkt het geluid natuurlijk absoluut onherkenbaar. De editor kan namelijk het geluid laten horen, zoals je het op dat moment hebt begrensd. Wel even opletten: wanneer je geluiden inkort en je verlaat de editor, ben je alle geluid buiten de grenzen die je had gesteld, onherroepelijk kwijt.

Terug naar het hoofdmenu via "Quit", en dan "Drums". Een set van acht drums, die elk één van de meegeleverde geluiden spelen. Ook hier dezelfde opmerking over de lay out van het toetsenbord.



Eventjes "ecHo" uitproberen. Blijkt dat ik niet voldoende geheugen meer over heb om een hoorbaar effect te produceren. Dan "dElete", twee geluiden wissen, en terug naar "ecHo". De vertraging is instelbaar. Het programma is intelligent : je kunt geen geluid overschrijven door de echotijd te lang te maken.

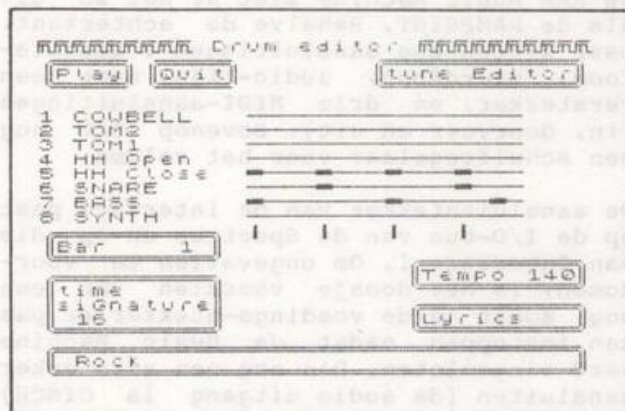
Het plezierigste vond ik het samplen zelf. De meegeleverde microfoon is niet bepaald de luxe, maar goed genoeg voor het experiment. De optie "sampleR" werkt simpel maar efficiënt. Je hebt de keuze tussen automatisch opstarten, of met de hand. Je kunt op een dB-schaal instellen bij welk volume de automaat begint te samplen. Met de microfoon is het mits wat geprobeer met de schuifregelaar best mogelijk om relatief zuivere stalen te nemen. Een piano klonk ook echt als een piano. Een viool was nauwelijks van echt te onderscheiden (nu ja...).



Het gemakkelijke is, dat je niet zo precies hoeft te samplen : er is altijd de editor, waarmee je exact die toon kan uitsnijden die je wenst, tot op een half sinusgolfje nauwkeurig als dat moet. Probeer wel om het niveau zo dicht mogelijk bij het 0 dB-punt te laten komen als je sampelt : dan is de signaal/ruis-verhouding het beste. Ruis wordt meege-sampeld en het valt niet mee om die eruit te snijden.

Iets moeilijker vond ik (niet-musicus) de "Bar editor", waar je echt muziek mee kunt schrijven. Je krijgt twee notenbalken (melodie en begeleiding), in sol en in fa. Eerst bepaal je het tempo en hoeveel noten per maat je wilt; dan schrijf je. Met de cursortoetsen beweeg je naar de plaats waar je een noot wilt zetten. Dan druk je op SPACE. Denk eraan dat het een tweekanaals-synthesiser is: hij kan maar twee noten tegelijk produceren. Daar let het programma zelf op : probeer je om er 3 tegelijk te laten horen dan wordt de voorlaatste weggehaald van zodra je de derde schrijft.

Natuurlijk kan je ook verhoogde en verlaagde noten schrijven. Om een noot te verhogen, druk je "s" ("sharp"), ga je met de cursor naar de noot die je wilt bewerken en druk je de spatietoets in. Dat geldt ook wanneer een noot verlaagd moet worden : druk op de "f" ("flat") en SPACE. Het programma verbetert fouten : wanneer je een do (C) wilt verlagen, wordt er een si (B) gezet. Slim !

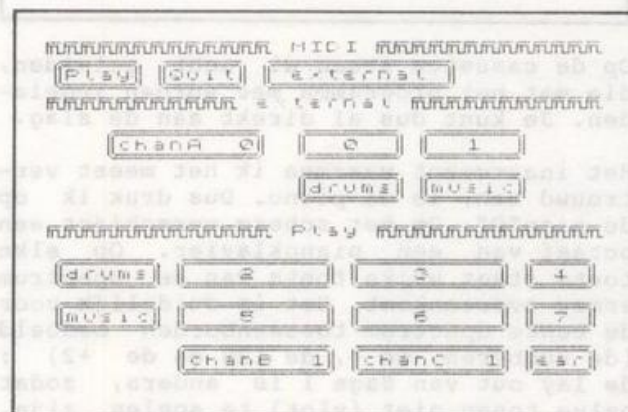


Zo schrijf je muziek, maat per maat. Je kunt elke maat een naam geven of een volgnummer. Om het "opus" samen te stellen, hoef je enkel via de "tuNe editor" een reeks van die namen of volgnummers in te geven. Hetzelfde doe je voor de ritmesectie : ook de drums kan je maat per maat schrijven, en later samenvoegen tot een volledige begeleiding. Druk je daarna op de "P", dan hoor je het hele stuk, ofwel de muziek ofwel de drums.

Je kunt natuurlijk ook geluid op band zetten voor later. Het programma is op een zeer eenvoudige manier op microdrive of Opus disk over te zetten.

Het kan gebeuren dat een geluid dat je sampelt, een andere klankkleur krijgt wanneer je het via de Music Machine laat horen (bv. een tik met een metalen voorwerp tegen een lege glazen fles). Door de sample-frekwentie van 19,5 kHz houd je nog een bandbreedte van 9,5 kHz over.

(lees verder op blz 46)



PROGRAMMA'S VAN LEZERS

In deze rubriek zult u in elk nummer een aantal programma's vinden, die door lezers werden ingezonden. De auteur(s) van elk gepubliceerd programma zullen wij natuurlijk belonen, want tenslotte geldt nog steeds: "Voor wat, hoort wat". Afhankelijk van de lengte van het programma, ontvangt de inzender een boekenbon van fl.15, fl.25 of fl.40. We proberen zoveel mogelijk programma's te publiceren. Kwaliteit speelt uiteraard de belangrijkste rol.

HOE PROGRAMMA INZENDEN

Houd er rekening mee dat wij echt geen tijd hebben om alle programma's die we toegestuurd krijgen, ook nog zelf in te typen en uit te zoeken hoe ze werken. Dat is echt niet mogelijk. Uitzondering kan gemaakt worden voor zeer korte dingetjes. Daarom vragen wij om programma's op te sturen op een cassette, microdrive-cartridge of Opus diskette. Die krijgt u na gebruik weer terug.

Het zou handig zijn, als we ook wisten waarvoor het programma bedoeld is en hoe het bediend moet worden, en waarop we moeten letten bij het save en load enzovoort. Probeer u om die uitleg kort en duidelijk te houden. Denk eraan, dat het programma alleen voor uzelf doorzichtig is, niet voor iedereen. Vergeet niet om instructies te geven in verband met SAVE en LOAD-volgorde, indien het programma uit een aantal onderdelen bestaat. Wat UDG's en blok-graphics betreft: onze printers verteren die niet. Die tekens hebben een code tussen 127 en 165. Gebruik die in de listing. Dat is niet zo mooi, maar wel duidelijk.

Het is prettig voor ons, dat ook de bijbehorende tekst (41 tekens breed getypt) op dezelfde drager staat. Tasword is onderhand wel zo verspreid, dat iedereen er een heeft of kan lenen. Andere systemen kunnen ook, maar wel graag zo, dat we niet de hele tekst zelf moeten intypen. Voor QL-teksten is een Quill-file natuurlijk evenzeer welkom.

Wat Assembler-listings betreft, nog even dit. Wij hebben niet alle bestaande Assemblers op ons kantoor, waardoor we met Assembler source-files niet altijd goed weg kunnen. Voor de Spectrum hebben we de Devpac en de Picturesque; voor de QL hebben we de Macro Assembler van GST.

Een listing op een margrietwiel-printer is zeer welkom. Ook doenbaar is: een listing in double-strike kwaliteit met een recent lint op uw printer, zodanig dat er een leesbare afdruk van gemaakt kan worden. Anders krijgen wij namelijk last met onze drukker...

Aan zeer actieve lezers, het verzoek om ons alstublieft geen boekwerken toe te zenden. Wij begrijpen dat sommige programma's nu eenmaal niet korter kunnen, en dat bij een geniale truuk wel een REMmetje mag. Maar op de ZX81 na kunnen alle huidige Sinclairs echt meer dan 1 statement op een regel bevatten. Houd kommentaar in een programma kort. Lange REMs in een listing maken hem onduidelijker - precies het tegenovergestelde van de bedoeling van een REMark.

WELKE PROGRAMMA'S

Wij kunnen onmogelijk zelf controleren of alle programma's echt wel eigen werk zijn. Wij moeten wat dat betreft, op uw eerlijkheid vertrouwen. Bedenk evenwel dat "uw" programma wordt gezien door zeer velen, die ook nog wel andere tijdschriften lezen. En er is altijd wel iemand, die ook het tijdschrift heeft gelezen waarin "uw" programma stond. Al brengen de kraaien het uit ...

Wees eerlijk. Gebruik je een routine die niet van jezelf is, vermeld dat er dan bij. Wij komen het hoe dan ook te weten en we beloven dat we de "dader" met naam en toenaam publiceren. Schande is dan uw deel, en terecht. Niet doen, dus.

||| spectrum |||

SCHAATSEN WK 1987

Met de Wereldkampioenschappen Allround voor de deur kregen we van de heer Peters uit Almelo een tweetal programma's waarmee je op een snelle manier de WK van afstand tot afstand op de voet kunt blijven volgen.

Het eerste programma is het hoofdprogramma en het tweede is bedoeld om een klassement van de laatste 16 te maken. Het programma werkt met microdrive en dus ook met OPUS. Met cassette zou het programma ook moeten kunnen werken, maar dan zit U met het probleem van lange laadtijden en het vooruit- en terugspoelen van een speciale DATA-cassette.

U moet dan uiteraard de vele SAVE en LOAD commando's aanpassen.

Het hoofdprogramma werkt met een printer, om de resultaten per afstand en klassen af te drukken. In de hierbij gepubliceerde listing zijn alle regels, die iets met de printer doen geREMD. DeHr. Peters werkt met een seriele printer gezien het FORMAT "t" statement. Voor een Parallel-printer kunt U deze statements vervangen door LPRINT. Let er op dat alleen STREAM #4 iets afdrukt. STREAM#5, dient om printer codes van de seriele printer in te stellen.

Werken met het programma:

Na de start van het programma, wordt er direct gevraagd of er al een deelnemerslijst is. Hier toetst U de eerste keer [N] in. Daarna vult U de afstanden van het tournooi in: resp. 500, 5000, 1500 en 10000. Daarna komt U in het HOOFDMENU. Hier kiest U optie 7 "Correctie deelnemerslijst". Dan komt U in een SUBMENU en hier kiest U optie 1: "Vernieuwen". Dan pakt U Uw krant (Welke krant zal niet de deelnemerslijst van de WK publiceren?) en U vult alle deelnemers in samen met land van de deelnemer. Om te stoppen met invoeren voert U na de laatste deelnemer "XXX" in om te stoppen met invoer. Bij een oneven aantal deelnemers moet er 1 rijder alleen over de baan, maak daarom een extra deelnemer genaamd: "DUMMY" en geef deze een extra lange tijd bijv. 10.0 op de 500 meter, zodat "DUMMY" achteraan in het klassement komt te zitten. Daarna SAVEt U de deelnemerslijst met optie 4: "Opbergen op tape".

Nu Uw deelnemerslijst veilig op cartridge staat is het beste om eerst nog een deelnemerslijst via de printer af te drukken. Dit doen we omdat we bij het invullen van de gegevens het nummer van de deelnemer moeten invullen. Het programma zoekt daar dan zelf de naam bij.

We starten nu met het ingeven van de gegevens van de 500 meter. U had al eerder moeten aangeven welke afstand aan de beurt was en bij het begin van het tournooi is dat de 500 meter. U kunt de gegevens nu per paar invoeren. Dat gaat als volgt: Uit het HOOFDMENU kiest U optie 1: "Invoeren van rijders". Als U het eerste paar heeft ingevoerd komt U terug in het Hoofdmenu en dan kiest U optie 2: "Invoeren van tijden" en als U dit gedaan heeft kunt U verder met het volgende paar enz. Zit de 500 meter er op dan kiest U de optie om alles op te bergen op tape. Als dit gebeurd is, krijgt U de keuze [D]oorgaan of [S]toppen en omdat er nogal wat tijd zit tussen twee ritten kunt U het beste [S] kiezen. Als U dan later weer start met RUN kunt U de deelnemerslijst weer inladen en begint U aan de 5000 meter.

BELANGRIJK: Bij het invoeren van tijden, moet U dit altijd doen in minuten. Voorbeeld: Hein Vergeer raffelt de 500 meter af in 38.9 sec. Dit voert U in als: 0.389

Bij de andere afstanden geldt hetzelfde:
5000: 6 min. 50.9 sec. wordt: 6.509 !
1500: 1 min. 55.4 sec. wordt: 1.554 !
10000: 14 min. 58.3 sec. wordt: 14.583 !

Bij elke volgende afstand na de 500 meter wordt van de twee rijders ook het onderlinge verschil op de volgende afstand aangegeven. U heeft met deze twee programma's een uitstekend middel om de gehele WK op de voet te volgen en het is te hopen dat de NOS ook zoiets gaat gebruiken om de kijkers beter te informeren. U bent dankzij de heer Peters in ieder geval wel goed geïnformeerd.

Veel schaatsplezier !

HOOFDPROGRAMMA

```
1 REM Peters, Almelo
10 REM FORMAT "t";2400: OPEN #4;"t": O
PEN #5;"b"
20 PRINT AT 11,0;"Is er een deelnemers
lijst?": PRINT AT 12,5;"Ja/Nee": GO SU
B 600
30 IF I$="n" OR I$="N" THEN GO TO 50
40 CLS : PRINT AT 11,4;"Inlezen deelne
merslijst!": LOAD "*"m";1;"Naamlijst" DAT
A D$( )
45 PRINT AT 12,4;"Inlezen landenlijst"
: LOAD "*"m";1;"Landlijst" DATA e$( )
50 BEEP 0.3,20: CLS
60 PRINT AT 1,2;"Voer de afstanden in
volgorde": PRINT AT 2,2;"van het tournoo
i in!"
70 PRINT AT 5,5;"1-e afstand:": PRINT
AT 6,5;"2-e afstand:": PRINT AT 7,5;"3-e
afstand:": PRINT AT 8,5;"4-e afstand:":
DIM x(4)
80 FOR i=1 TO 4: INPUT "Afstand in met
ers: ";x(i): PRINT AT (4+i),20;x(i): NE
XT i
90 PRINT AT 15,5;"Invoer juist?": PRIN
T AT 17,10;"Ja/Nee": GO SUB 600
100 IF I$="n" OR I$="N" THEN GO TO 60
110 PRINT AT 15,02;"Welke afstand is aa
n de beurt?": PRINT AT 17,6;"Maak een ke
uze (1-4)"
120 GO SUB 600: LET k=VAL I$: IF k<1 OR
k>4 THEN GO TO 120
130 CLS
140 PRINT " ";: FOR i=1 TO (18+LEN
(STR$ (x(k)))): PRINT "*";: NEXT i: PRIN
T
150 PRINT " * Afstand ";x(k);" mete
r *"
160 PRINT " ";: FOR i=1 TO (18+LEN
(STR$ (x(k)))): PRINT "*";: NEXT i: PRIN
T
```



```

170 LET delen=x(k)/500: LET paar=1: LET
rijden=0: LET aantal=0
180 IF k=4 THEN LET maal=1: GO TO 200
190 LET maal=x(k+1)/500
200 DIM R$(45,10): DIM K$(45,10): DIM H(45)
(45): DIM L(45): DIM H$(45,10): DIM H(45
): DIM M$(2,10): DIM M(2): DIM Q$(1,10):
DIM Q(2): DIM A$(2,10): DIM A(2): DIM P
$(1,5): DIM V$(1,10): DIM W$(1,10)
210 REM PRINT #5;CHR$ 27;CHR$ 14;: REM
Begin uitrekken
220 REM PRINT #5;CHR$ 27;CHR$ 21;: REM
CR
230 REM PRINT #4;"      Rittenoverzicht!"
"
240 REM PRINT #4;"      ====="
250 REM PRINT #4
260 REM PRINT #4;"      Afstand: ";x(k);
" meter."
270 REM PRINT #4;"      Rit no :";k
280 REM PRINT #4
290 REM PRINT #5;CHR$ 27;CHR$ 15;: REM
Einde uitrekken
300 REM PRINT #4;"      Nr  Naam
Tijd      Punten      Puntentotaal": PRIN
T #4
310 CLS
320 IF k=1 THEN GO TO 370
330 LET Z$="tot$"+STR$ (x(k-1))
340 CLS : PRINT AT 0,2;"Inlezen klassem
ent ";x(k-1);" meter!"
350 PRINT AT 2,8;"Namen": LOAD $"m";1;Z
$ DATA T$( )
360 LET Z$="tot$"+STR$ (x(k-1)): PRINT A
T 4,8;"Punten": LOAD $"m";1;Z$ DATA T( )
370 BEEP 0.5,20
380
390
400 CLS : PRINT AT 0,5;"*Afstand ";x(k)
;" meter*"
410 PRINT "      ": FOR i=1 TO (16+LEN
(STR$ (x(k)))): PRINT "=";: NEXT i: PRIN
T
420 PRINT AT 4,2;"1 Invoer van rijders
:"
430 PRINT AT 6,2;"2 Invoer van tijden
:"
440 PRINT AT 8,2;"3 Rittenoverzicht
:"
450 PRINT AT 10,2;"4 Afstandsklassement
:"
460 PRINT AT 12,2;"5 Totaalklassement
:"
470 PRINT AT 14,2;"6 Resultaten op tape
:"
480 PRINT AT 16,2;"7 Correctie deelname
rsljst :":
490 PRINT AT 20,2;"Maak een keuze (1-7)
": GO SUB 600: LET I=VAL I$: IF I<1 OR I
>7 THEN GO TO 490
500 CLS : GO TO (I+1)*1000
510
600 IF INKEY$<>" " THEN GO TO 600
610 IF INKEY$="" THEN GO TO 610
620 LET I$=INKEY$: RETURN
630
2000 IF rijden=1 THEN GO TO 2210
2010 PRINT AT 2,10;"Rit no :";k
2020 PRINT AT 4,10;"Afstand: ";x(k);" me
ter"
2030 PRINT AT 6,10;"Paar no :";paar
2040 PRINT AT 10,6;"Wie rijdt tegen wie?"

```

```

2060 PRINT AT 14,4;"nr 1= ": PRINT AT 16
,4;"nr 2= "
2070 INPUT "Nr van 1-e rijder: (uit tabe
l)";i
2080 IF D$(I)="      " THEN GO SUB 2
250
2090 LET M$(1)=D$(I): PRINT AT 14,13;M$(
1)
2100 INPUT "Nr van 2-e rijder: (uit tabe
l)";i
2110 IF D$(I)="      " THEN GO SUB 2
250
2120 LET M$(2)=D$(I): PRINT AT 16,13;M$(
2)
2130 PRINT AT 18,6;"Invoer juist??"
2140 PRINT AT 20,8;"Ja/nee": GO SUB 600
2150 IF I$="n" OR I$="N" THEN GO TO 2000
2160 LET rijden=1: LET aantal=aantal+2:
LET R$(aantal-1)=M$(1): LET R$(aantal)=M
$(2)
2170 PRINT AT 18,6;M$(1);"      ": PRINT
AT 20,6;M$(2)
2180 LET Q$(1)=M$(1): GO SUB 2300: LET Q
(1)=j: PRINT AT 18,18;Q(1)
2190 LET Q$(1)=M$(2): GO SUB 2300: LET Q
(2)=j: PRINT AT 20,18;Q(2)
2200 PRINT #0;"Druk op toets": PAUSE 0:
GO TO 400
2210 CLS : BEEP 1,20: PRINT AT 11,5;"Paa
r no ";paar;" is nog bezig": PRINT AT 13
,5;"Druk op toets": PAUSE 0: GO TO 400
2220
2250 PRINT AT 18,6;"Naam van rijder": IN
PUT D$(I): RETURN
2260
2300 IF k=1 THEN LET j=0: RETURN
2310 LET i=0
2320 LET i=i+1: IF i>45 THEN GO TO 2350
2330 IF Q$(1)<>T$(i) THEN GO TO 2320
2340 LET j=T(i): RETURN
2350 LET j=300: RETURN : REM onbekend
2360
3000 IF rijden=0 THEN PRINT AT 11,5;"Tij
d is al ingevoerd": BEEP 1,20: GO TO 400
3010 PRINT AT 1,8;"Rit no :";k
3020 PRINT AT 2,8;"Paar no: ";paar
3030 PRINT AT 3,8;"Afstand: ";x(k);" mete
r."
3040 PRINT AT 5,4;M$(1): PRINT AT 5,16;"
": PRINT AT 5,18;M$(2)
3050 PRINT AT 7,5; FLASH 1;"1": PRINT AT
7,21; FLASH 1;"2"
3060 PRINT AT 10,3;"Wie is winnaar: 1 of
2 ??": GO SUB 600
3070 LET I=VAL I$: IF I<1 OR I>2 THEN GO
TO 3050
3080 PRINT AT 12,0;"nr": PRINT AT 12,3;"
Naam": PRINT AT 12,15;"Tijd": PRINT AT 1
2,23;"Punten"
3090 INPUT "Winnende tijd: ";tijd$
3100 LET a=INT (tijd$): LET b=(tijd$-a)*
100
3110 LET a=(a*60+b)/delen
3120 LET C=I: LET M(C)=INT (a*1000+0.5)/
1000: LET A(1)=M(C)+Q(C): LET A$(1)=M$(C
)
3130 PRINT AT 14,0;"1": PRINT AT 14,3;M$(
C): PRINT AT 14,15;tijd$: PRINT AT 14,2
3;M(C)
3140 LET I=I+1
3150 IF I=3 THEN LET I=1
3160 INPUT "Tijd van 2-e rijder: ";tijdy
3170 LET a=INT (tijdy): LET b=(tijdy-a)*
100

```



```

3180 LET a=(a*60+b)/delen
3190 LET D=I: LET M(D)=INT (a*1000+0.5)/
1000: LET A(2)=M(D)+Q(D): LET A$(2)=M$(D)
)
3200 PRINT AT 16,0;"2": PRINT AT 16,3;M$(
D): PRINT AT 16,15;tijdy: PRINT AT 16,2
3;M(D)
3210 PRINT AT 18,5;"Invoer juist?": PRIN
T AT 20,10;"Ja/Nee ": GO SUB 600
3220 IF I$="n" OR I$="N" THEN CLS : GO T
O 3000
3230 REM PRINT #5;CHR$ 27;CHR$ 54;
3240 REM PRINT #5;CHR$ 27;CHR$ 21;
3250
3260 LET P$(1)=STR$ (aantal-1): LET V$(1
)=STR$ (tijdx): LET W$(1)=STR$ M(C)
3270 REM PRINT #4;" " ;P$(1);M$(C);"
";V$(1);W$(1);A(1)
3280 LET P$(1)=STR$ (aantal): LET V$(1)=
STR$ (tijdy): LET W$(1)=STR$ M(D)
3290 REM PRINT #4;" " ;p$(1);M$(D);"
";V$(1);W$(1);A(2): PRINT #4
3300 IF aantal>2 THEN GO TO 3370
3310 LET K$(1)=M$(C): LET K(1)=tijdx: LE
T L(1)=M(C)
3320 LET K$(2)=M$(D): LET K(2)=tijdy: LE
T L(2)=M(D)
3330 LET H$(1)=A$(1): LET H$(2)=A$(2): L
ET H(1)=A(1): LET H(2)=A(2)
3340 IF H(1)<H(2) THEN GO TO 3370
3350 LET Q$(1)=H$(1): LET Q(1)=H(1): LET
H$(1)=H$(2): LET H(1)=H(2): LET H$(2)=Q
$(1): LET H(2)=Q(1)
3360 GO TO 3610
3370 PRINT AT 14,0; BRIGHT 1;"1"
3380 LET I=0
3390 LET I=I+1
3400 IF M(C)>L(I) AND L(I)<>0 THEN GO TO
3390
3410 FOR J=(aantal-1) TO (I+1) STEP -1:
LET L(J)=L(J-1): LET K$(J)=K$(J-1): LET
K(J)=K(J-1): NEXT J
3420 LET K$(I)=M$(C): LET K(I)=tijdx: LE
T L(I)=M(C)
3430 PRINT AT 14,0;"1": PRINT AT 16,0; B
RIGHT 1;"2"
3440 LET I=I+1
3450 IF M(D)>L(I) AND L(I)<>0 THEN GO TO
3440
3460 FOR J=aantal TO I STEP -1: LET L(J)
=L(J-1): LET K$(J)=K$(J-1): LET K(J)=K(J
-1): NEXT J
3470 LET K$(I)=M$(D): LET K(I)=tijdy: LE
T L(I)=M(D)
3480 IF A(1)<A(2) THEN GO TO 3500
3490 LET Q$(1)=A$(1): LET Q(1)=A(1): LET
A$(1)=A$(2): LET A(1)=A(2): LET A$(2)=Q
$(1): LET A(2)=Q(1)
3500 PRINT AT 14,0; BRIGHT 1;">": PRINT
AT 16,0;"2"
3510 LET I=0
3520 LET I=I+1
3530 IF A(1)>H(I) AND H(I)<>0 THEN GO TO
3520
3540 FOR J=(aantal-1) TO (I+1) STEP -1:
LET H(J)=H(J-1): LET H$(J)=H$(J-1): NEXT
J
3550 LET H$(I)=A$(1): LET H(I)=A(1)
3560 PRINT AT 14,0;">": PRINT AT 16,0; B
RIGHT 1;">"
3570 LET I=I+1

```

```

3580 IF A(2)>H(I) AND H(I)<>0 THEN GO TO
3570
3590 FOR J=aantal TO I STEP -1: LET H$(J)
=H$(J-1): LET H(J)=H(J-1): NEXT J
3600 LET H$(I)=A$(2): LET H(I)=A(2)
3610 LET rijden=0: LET paar=paar+1: BEEP
0.3,20: GO TO 400
3620
4000 CLS : PRINT AT 0,3;"Rittenoverzicht
";x(k);" meter"
4010 PRINT " " ;: FOR i=1 TO (22+LEN (S
TR$ (x(k)))): PRINT "=";: NEXT i: PRINT
4015 LET y=1
4020 FOR I=1 TO aantal STEP 2
4030 PRINT y;: PRINT TAB 4;R$(I);: PRINT
TAB 16;" ";: PRINT TAB 18;R$(I+1)
4035 LET y=y+1
4040 NEXT I
4050 PRINT #0;"Druk op toets": PAUSE 0:
GO TO 400
4060
5000 PRINT AT 0,3;"Klassement ";x(k);" m
eter"
5010 PRINT " " ;: FOR i=1 TO (17+LEN (S
TR$ (x(k)))): PRINT "=";: NEXT i: PRINT
5020 LET y=15: IF aantal<=15 THEN LET y=
aantal
5030 FOR i=1 TO y
5040 PRINT I;: PRINT TAB 4;k$(i);: PRINT
TAB 15;k(i);: PRINT TAB 24;l(i)
5050 NEXT i
5060 PRINT #0;"Druk op toets": PAUSE 0
5070 IF aantal>15 THEN CLS : GO TO 5090
5080 GO TO 400
5090 PRINT AT 0,3;"Klassement ";x(k);" m
eter"
5100 PRINT " " ;: FOR i=1 TO (17+LEN (S
TR$ (x(k)))): PRINT "=";: NEXT i: PRINT
5110 LET y=30: IF aantal<=30 THEN LET y=
aantal
5120 FOR i=16 TO y
5130 PRINT I;: PRINT TAB 4;k$(i);: PRINT
TAB 15;k(i);: PRINT TAB 24;l(i)
5140 NEXT i
5150 PRINT #0;"Druk op toets": PAUSE 0
5160 IF aantal>30 THEN CLS : GO TO 5180
5170 GO TO 400
5180 PRINT AT 0,3;"Klassement ";x(k);" m
eter"
5190 PRINT " " ;: FOR i=1 TO (17+LEN (S
TR$ (x(k)))): PRINT "=";: NEXT i: PRINT
5200 LET y=45: IF aantal<=45 THEN LET y=
aantal
5210 FOR i=31 TO y
5220 PRINT I;: PRINT TAB 4;k$(i);: PRINT
TAB 15;k(i);: PRINT TAB 24;l(i)
5230 NEXT i
5240 PRINT #0;"Druk op toets": PAUSE 0:
GO TO 400
5250
6000 IF k=4 THEN LET z$="( " ;: GO T
O 6020
6010 LET z$="("+STR$ (x(k+1))+")"
6020 PRINT AT 0,3;"Totaalklassement ";x(
k);" meter"
6030 PRINT " " ;: FOR i=1 TO (23+LEN (S
TR$ (x(k)))): PRINT "=";: NEXT i: PRINT
6040 PRINT "Nr Naam Punten ";z$
6050 LET y=15: IF aantal<=15 THEN LET y=
aantal
6060 FOR i=1 TO y

```



```

6070 LET p$(1)=STR$(INT(((H(I)-H(1))*m
aal)*1000)/1000)
6080 PRINT I; PRINT TAB 4;H$(i); PRINT
TAB 16;H(i); PRINT TAB 24;p$(1)
6090 NEXT i
6100 PRINT #0;"Druk op toets ": PAUSE 0
6110 IF aantal>15 THEN CLS : GO TO 6130
6120 GO TO 400
6130 PRINT AT 0,3;"Totaalklassement ";x(
k);" meter"
6140 PRINT " ";: FOR i=1 TO (23+LEN(S
TR$(x(k)))): PRINT "=";: NEXT i: PRINT
6150 PRINT "Nr Naam Punten ";z$
6160 LET y=30: IF aantal<=30 THEN LET y=
aantal
6170 FOR i=16 TO y
6180 LET p$(1)=STR$(INT(((H(I)-H(1))*m
aal)*1000)/1000)
6190 PRINT I; PRINT TAB 4;H$(i); PRINT
TAB 16;H(i); PRINT TAB 24;p$(1)
6200 NEXT i
6210 PRINT #0;"Druk op toets ": PAUSE 0
6220 IF aantal>30 THEN CLS : GO TO 6240
6230 GO TO 400
6240 PRINT AT 0,3;"Totaalklassement ";x(
k);" meter"
6250 PRINT " ";: FOR i=1 TO (23+LEN(S
TR$(x(k)))): PRINT "=";: NEXT i: PRINT
6260 PRINT "Nr Naam Punten Ve
rschil (";z$;")"
6270 LET y=45: IF aantal<=45 THEN LET y=
aantal
6280 FOR i=31 TO y
6290 LET p$(1)=STR$(INT(((H(I)-H(1))*m
aal)*1000)/1000)
6300 PRINT I; PRINT TAB 4;H$(i); PRINT
TAB 16;H(i); PRINT TAB 24;p$(1)
6310 NEXT i
6320 PRINT #0;"Druk op toets": PAUSE 0:
GO TO 400
6330
6990 REM Wegschrijven van resultaten
7000 PRINT AT 1,3;"Wegschrijven van resu
ltaten": PRINT AT 2,3;"*****"
7020 PRINT AT 4,1;"Afstandsklassement ";
x(k);" meter"
7030 PRINT AT 6,3;"Namen": PRINT AT 8,3;
"Tijden": PRINT AT 10,3;"Punten"
7040 IF k=1 THEN LET z$="rit": GO TO 705
0
7045 LET z$=" ritten"
7050 PRINT AT 14,3;"Totaalklassement na
";k;z$
7060 PRINT AT 16,3;"Namen": PRINT AT 18,
3;"Punten"
7070 PRINT AT 6,3; BRIGHT 1;"Namen";x(k)
;" meter": LET z$="naam"+STR$(x(k)): SAVE
"m";1;z$ DATA k$()
7080 PRINT AT 8,3; BRIGHT 1;"Tijden";x(k)
;" meter": LET z$="tijd"+STR$(x(k)): SAV
E "m";1;z$ DATA k$()
7090 PRINT AT 10,3; BRIGHT 1;"Punten";x(
k);" meter": LET z$="punt"+STR$(x(k)): SA
VE "m";1;z$ DATA k$()
7100 PRINT AT 16,3; BRIGHT 1;"Namen";x(k)
;" meter": LET z$="tot"+STR$(x(k)): SAV
E "m";1;z$ DATA h$()
7110 PRINT AT 18,3; BRIGHT 1;"Punten";x(
k);" meter": LET z$="tot"+STR$(x(k)): SAV
E "m";1;z$ DATA h$()
7120 BEEP 1,20: PRINT #0;"Zet printer go

```

```

ed ": PAUSE 0
7130 LET z$="ritten": DIM y$(1,5)
7140 IF k=1 THEN LET z$="rit"
7150 REM PRINT #5;CHR$(27);CHR$(14);
7160 REM PRINT #5;CHR$(27);CHR$(21);
7170 REM PRINT #4;" Klassement na ";k
;" ";z$
7180
7190 REM PRINT #5;CHR$(27);CHR$(15);
7195 REM PRINT #5;CHR$(27);CHR$(54);
7200 REM PRINT #4;"Nr Naam Punten
";: PRINT #4
7210 FOR i=1 TO aantal
7220 LET p$(1)=STR$(i): LET y$(1)=STR$(
INT(((H(I)-H(1))*maal)*1000)/1000)
7230 REM PRINT #4;p$(1);h$(i);h(i);" "
;y$(1)
7240 NEXT i
7920 PRINT #0;"Doorgaan of stoppen? D/S?
??"
7930 GO SUB 600: IF I$="s" OR I$="S" THE
N GO TO 7950
7940 LET k=k+1: GO TO 130
7950 CLS
7960 PRINT AT 10,2;"Hartelijk dank voor
uw": PRINT AT 11,2;"welwillende medewerk
ing.": PRINT AT 12,2;"Onze samenwerking
was": PRINT AT 13,2;"uiterst frappant!!"
": PRINT AT 15,5;"TOT ZIENS!"
7970 REM CLOSE #4: CLOSE #5: STOP
8000 PRINT AT 0,3;"Deelnemerslijst"
8010 PRINT AT 1,3;"=====
8020 PRINT AT 3,5;"1 Vernieuwen": PRINT
AT 5,5;"2 Aanvullen": PRINT AT 7,5;"3 Wi
jzigen": PRINT AT 9,5;"4 Opbergen op tap
e": PRINT AT 11,5;"5 Terug naar menu"
8030 PRINT AT 15,5;"Maak een keuze (1-5)
"
8040 GO SUB 600: IF VAL I$<1 OR VAL I$>5
THEN GO TO 8040
8050 IF VAL I$=1 THEN GO TO 8450
8060 IF VAL I$=2 OR VAL I$=3 THEN PRINT
AT 17,5;"Even wachten a.u.b.": GO TO 808
0
8070 IF VAL I$=4 THEN GO TO 8410
8075 IF VAL I$=5 THEN GO TO 400
8080 LET q$(1)=" "
8090 LET a=0
8100 LET a=a+1: IF a=46 THEN GO TO 8120
8110 IF d$(a)<>q$(1) THEN GO TO 8100
8120 CLS : LET a=a-1
8130 PRINT AT 0,3;"Deelnemerslijst"
8140 PRINT AT 1,3;"=====
8150 LET y=15: IF a<=15 THEN LET y=a
8160 FOR j=1 TO y: PRINT j;: PRINT TAB 4
;d$(j);: PRINT TAB 16;e$(j): NEXT j
8170 PRINT #0;"Druk op toets": PAUSE 0
8180 IF a>15 THEN CLS : GO TO 8200
8190 IF VAL I$=1 THEN GO TO 8000
8195 IF VAL I$=3 THEN GO TO 8370
8200 LET y=30: IF a<=30 THEN LET y=a
8210 FOR j=16 TO y: PRINT j;: PRINT TAB
4;d$(j);: PRINT TAB 16;e$(j): NEXT j
8220 PRINT #0;"Druk op toets": PAUSE 0
8230 IF a>30 THEN CLS : GO TO 8250
8240 IF VAL I$=1 THEN GO TO 8000
8245 IF VAL I$=3 THEN GO TO 8370
8250 LET y=45: IF a<=45 THEN LET y=a
8260 FOR j=31 TO y: PRINT j;: PRINT TAB
4;d$(j);: PRINT TAB 16;e$(j): NEXT j
8270 PRINT #0;"Druk op toets": PAUSE 0
8280 IF VAL I$=1 THEN GO TO 8000

```



```

8290 IF VAL I$=3 THEN GO TO 8370
8300 LET a=a+1: IF a=46 THEN PRINT "Lijst is vol": BEEP 0.5,20: PRINT #0;"Druk op toets": PAUSE 0: GO TO 8000
8310 INPUT "Naam van rijder";d$(a): INPUT "Land";e$(a)
8320 PRINT #0;"Nog meer? "
8330 GO SUB 600: IF I$="j" OR I$="J" THEN GO TO 8300
8340 CLS : GO TO 8000
8350
8360
8370 INPUT "Welk nr wijzigen? ";a: INPUT "Nieuwe naam:";d$(a): INPUT "Land:";e$(a)
8380 PRINT #0;"Nog meer wijzigen? "
8390 GO SUB 600: IF I$="j" OR I$="J" THEN GO TO 8370
8400 CLS : GO TO 8000
8402
8405 REM Opbergen op tape
8408
8410 CLS : PRINT AT 11,0;"Is er al een deelnemerslijst?": PRINT AT 13,10;"Ja/Nee"
8420 GO SUB 600: IF I$="n" OR I$="N" THEN GO TO 8435
8430 PRINT AT 15,10; BRIGHT 1;"Wissen": ERASE "m";1;"Naamlijst": ERASE "m";1;"Landlijst"
8435 PRINT AT 16,10; BRIGHT 1;"Saven": SAVE "m";1;"Naamlijst" DATA d$(): SAVE "m";1;"Landlijst" DATA e$()
8440 GO TO 400
8450 CLS : PRINT AT 0,3;"Deelnemerslijst (nieuw)": PRINT AT 1,3;"=====
=====
8460 DIM d$(45,10): DIM e$(45,5)
8470 LET a=0
8480 LET a=a+1
8490 INPUT "Naam van rijder: ";z$
8500 IF z$="xxx" OR z$="XXX" THEN GO TO 8120
8510 LET d$(a)=z$: INPUT "Land:";e$(a)
8520 PRINT a;: PRINT TAB 4;d$(a);: PRINT TAB 16;e$(a): GO TO 8480
8530 STOP
9000 REM PRINT #5;CHR$ 27;CHR$ 14;
9010 FOR i=1 TO 45
9020 LET p$(1)=STR$(i)
9030 REM PRINT #4;p$(1);d$(i);" ";e$(i)
9040 NEXT i
9050 REM PRINT #5;CHR$ 27;CHR$ 15;

```

LAATSTE 16

```

10 REM FORMAT "t";2400: OPEN #4;"t": OPEN #5;"b"
15 INPUT "Voer afstanden in ";a1,a2,a3,a4
20 DATA a1,a2,a3,a4
30 DIM a$(1,2): DIM b$(16,3): DIM c$(1,7): DIM d$(1,5): DIM e$(1,5): DIM r$(1,2): DIM f$(1,6): DIM g$(1,7): DIM s$(1,2): DIM h$(1,6): DIM k$(1,7): DIM t$(1,2): DIM l$(1,7): DIM u$(1,2)
40 DIM c(16): DIM d(16): DIM e(16): DIM r(16): DIM f(16): DIM g(16): DIM s(16): DIM h(16): DIM k(16): DIM t(16): DIM l(16): DIM u(16)

```

```

50 PRINT AT 8,5;"Inlezen namenlijst": PRINT AT 10,7;"en landenlijst"
60 LOAD "m";1;"Naamlijst" DATA x$(): LOAD "m";1;"Landlijst" DATA y$( )
70 BEEP 0.5,20: CLS : PRINT AT 8,5;"Samenstellen van namen": PRINT AT 10,5;"en bijbehorend land.": PRINT AT 12,5;"Tevens puntentotaal": BEEP 0.5,20
80 PRINT AT 14,5; BRIGHT 1; FLASH 1;"LADEN": LOAD "m";1;"tot$10000" DATA z$(): PRINT AT 14,22; BRIGHT 1; FLASH 1;"LADEN": LOAD "m";1;"tot$10000" DATA z$()
85 CLS : PRINT AT 10,4;"Lijsten combineren!"
90 FOR i=1 TO 16
100 LET j=0
110 LET j=j+1
120 IF x$(j)<>z$(i) THEN GO TO 110
130 LET b$(i)=y$(j)
140 NEXT i
150 BEEP 1,20: CLS
160 FOR i=1 TO 16
170 PRINT TAB 3;i;: PRINT TAB 6;z$(i);: PRINT TAB 18;b$(i);: PRINT TAB 22;z(i)
180 NEXT i
190 BEEP 1,20: PRINT #0;"Druk op toets": PAUSE 0
200 CLS : GO TO 240
210 CLS : READ afstand
220 PRINT AT 12,1;"Laden namen ";a: fstand;" meter": LOAD "m";1;"naam"+STR$(afstand) DATA p$(): PRINT AT 13,1;"Laden tijden ";a: fstand;" meter": LOAD "m";1;"tijd"+STR$(afstand) DATA p()
230 PRINT AT 14,1;"Laden totalen ";a: fstand;" meter": LOAD "m";1;"tot$"+STR$(afstand) DATA q$(): PRINT AT 15,1;"Laden punten na ";a: fstand;" meter": LOAD "m";1;"tot"+STR$(afstand) DATA q(): PRINT AT 17,1;"Gegevens verwerken!": BEEP 0.5,20: RETURN
240 GO SUB 210: REM Resultaat 500 meter
245 FOR i=1 TO 16
250 LET j=0
260 LET j=j+1
270 IF p$(j)<>z$(i) THEN GO TO 260
280 LET d(i)=p(j): LET r(i)=j
290 LET j=0
300 LET j=j+1
310 IF q$(j)<>z$(i) THEN GO TO 300
320 LET e(i)=q(j)
330 NEXT i
340 GO SUB 210: REM Resultaat 5000 meter
350 FOR i=1 TO 16
360 LET j=0
370 LET j=j+1
380 IF p$(j)<>z$(i) THEN GO TO 370
390 LET f(i)=p(j): LET s(i)=j
400 LET j=0
410 LET j=j+1
420 IF q$(j)<>z$(i) THEN GO TO 410
430 LET g(i)=q(j)
440 NEXT i
450 GO SUB 210: REM Resultaat 1500 meter
460 FOR i=1 TO 16
470 LET j=0
480 LET j=j+1
490 IF p$(j)<>z$(i) THEN GO TO 480
500 LET h(i)=p(j): LET t(i)=j
510 LET j=0

```



```

520 LET j=j+1
530 IF q$(j)<>z$(i) THEN GO TO 520
540 LET k(i)=q(j)
550 NEXT i
560 GO SUB 210: REM Resultaat 10000 met
er
570 FOR i=1 TO 16
580 LET j=0
590 LET j=j+1
600 IF p$(j)<>z$(i) THEN GO TO 590
610 LET l(i)=p(j): LET u(i)=j
620 NEXT i
630
640 REM PRINT #5;CHR$ 27;CHR$ 21;
650 REM PRINT #5;CHR$ 27;CHR$ 23;
660 FOR i=1 TO 16
670 LET a$(1)=STR$ (i): LET c$(1)=STR$
(z(i)): LET d$(1)=STR$ (d(i)): LET e$(1)
=STR$ (e(i)): LET r$(1)=STR$ (r(i)): LET
f$(1)=STR$ (f(i)): LET g$(1)=STR$ (g(i)
): LET s$(1)=STR$ (s(i)): LET h$(1)=STR$
(h(i)): LET k$(1)=STR$ (k(i)): LET t$(1)
=STR$ (t(i)): LET l$(1)=STR$ (l(i)): LE
T u$(1)=STR$ (u(i))
680 REM PRINT #4;a$(1);" ";z$(i);" ";b$
(i);" ";c$(1);" ";e$(1);" (";r$(1);" "
f$(1);" ";g$(1);" (";s$(1);" "h$(1);"
";k$(1);" (";t$(1);" "l$(1);" (";u$(1)
";"
690 NEXT i
700 REM CLOSE #4: CLOSE #5

```

LUXE LISTER

EBERT ZWART

Dit machinetaal programma zorgt ervoor dat uw BASIC listings op een overzichtelijke manier op uw printer verschijnen. Het heeft ten opzichte van de gebruikelijke LLIST een aantal voordelen:

1. KEYWORDS die niet meer op de regel passen, worden op de volgende regel geprint.
2. De eerste 4 posities van elke regel zijn gereserveerd voor het regelnummer, zodat de regelnummers goed zichtbaar zijn en er niet onder wordt doorgeschreven als de BASIC regel niet op de printregel past.
3. UDG's en grafische characters kunnen ook geprint worden. Zie verderop.
4. De linker- en rechterkantlijn kunnen naar eigen smaak ingevuld worden. Tevens kan de laatste printregel per blad worden opgegeven.
5. Op ieder nieuw blad wordt een kopregel geprint met de titel van het programma.

6. Er bestaat de mogelijkheid om een deel van het programma te listen.

Voor de keuzes 3 en 4 is een speciaal installeerprogramma bijgeleverd. De UDG karakters A t/m U worden op de printer ook als A t/m U geprint. U kunt echter een of meerdere control characters meegeven voor een UDG. U zou een UDG bijvoorbeeld schuin en onderstreept kunnen afdrukken. Net als voor het activeren van een UDG karaktersoort, moeten ook de control characters voor het deactiveren opgegeven worden. De grafische karakters kunnen ook afgedrukt worden mits uw printer beschikt over grafische mogelijkheden. Er moet dan een control character gestuurd worden om de printer in grafische mode te zetten. Bij de meeste printers moet nu de hoeveelheid te zenden bytes opgegeven worden. Deze hoeveelheid is acht. Ideale dot spacing is 1/72 inch omdat de verhouding omhoog/opzij dan 1:1 is. Heeft uw printer geen grafische mogelijkheden, dan wordt er een ? afgedrukt. De control characters die u kunt gebruiken, staan in uw printer handleiding. Het aantal control characters dat per optie meegegeven mag worden is 7.

De linker- en rechterkantlijn kunt u zelf bepalen, mits deze minimaal 32 karakters uit elkaar liggen. Ook kunt u de laatste printregel bepalen. Voor 12 inch papier is de adviesregel 68 en voor 11 inch papier 62.

Instructies voor saven, laden en runnen:

Type de 48K (of 16K) loader in en RUN het programma. Als u ergens een typfout hebt gemaakt, krijgt u een boodschap gevolgd door "STOP statement". U moet net zo lang de fouten in uw listing verbeteren totdat u tijdens de RUN de boodschap "Out of DATA" krijgt. Save de verkregen code nu op tape met SAVE "LISTER" CODE 64512,727 (of 16K: 31744,727) en spoel de tape terug. Typ nu de volgende commando's in:

```
CLEAR 31743
```

```
NEW
```

Typ nu het installeerprogramma in en RUN het. Aan het eind van de run wordt de geïnstalleerde machinecode gesaved. Dit is het uiteindelijke programma. Om het programma te gebruiken, moet u het in het geheugen laden samen met het te listen BASIC programma. Dit doet u dmV LOAD "LISTER" CODE.

Om deze BASIC lister vervolgens te kunnen gebruiken, hebt u twee kanalen nodig. De volgende statements initialiseren deze kanalen. Deze statements moet u dus los invoeren.

OPEN #3,"t": OPEN #4,"b"

Vervolgens moet u de baudrate van uw printer bekend maken met:

FORMAT "t";baudrate

De code, die voor 48K gebruikers op 64512 begint en voor 16K gebruikers op 31744, is 727 bytes lang en kan worden aangeroepen met RANDOMIZE USR 64512 (31744 voor 16K). In dat geval wordt de complete listing van het BASIC programma dat in het geheugen staat naar de printer gestuurd. Als u slechts een deel van het programma wilt uitprinten, dan moet er in het BASIC programma een regel staan met:

```
DEF FN A(f,1) = USR 64512 (48K)
DEF FN A(f,1) = USR 31744 (16K)
```

De deellisting kan dan afgedrukt worden met RANDOMIZE FN A(eerste,laatste). Hierbij vult u op de plaats van 'eerste' de beginregel in en op de plaats van 'laatste' de eindregel in.

De naam van het programma komt in de kopregel te staan, die op elke bladzijde verschijnt. Deze naam moet in de eerste regel van het programma in een REM statement zijn opgenomen. Hij moet 10 karakters lang zijn. Voorbeeld:

```
10 REM LISTERbbbb (b = spatie)
```

Als het eerste statement niet een REM statement is, worden er vraagtekens op de plaats van de naam geprint.

```
0 REM 48K LOADER
0 REM Gecreeerd met HEXLOAD
30
40 DEF FN F(X$)=16*(CODE X$(1)-48-39*(
X$(1)>"9"))+CODE X$(2)-48-39*(X$(2)>"9")
50 LET ADR=VAL "64512": LET L=1000
60 READ H$: LET SOM=0
70 FOR C=1 TO LEN (H$) STEP 2
80 LET WRD=FN F(H$(C TO C+1))
90 POKE ADR,WRD
100 LET ADR=ADR+1
110 LET SOM=SOM+WRD
120 NEXT C
130 READ TOTAAL
135 IF TOTAAL<>SOM THEN PRINT "FOUTE CH
EKSOM IN REGEL ";L: STOP
140 IF TOTAAL=SOM THEN PRINT "Regel ";L
;" OK"
145 LET L=L+10
150 GO TO 60
1000 DATA "1828031b2d01000000000031b2d000
0000000051b2a0508000000000000000000000
8484400000000002a535c23232323eeabe200b2
311cafe010a00edb018",VAL "2292"
1010 DATA "0a21cafe060a363f2310fb3efa322
8fcdf3229fc2a0b5c7cb5281ee5dde1dd6e0cdd6
60d23cd6e192226fcdd6e04dd6605cd6e192b3e0
d1812211027cd6e1922",VAL "6304"
```

```
1020 DATA "26fc2a535c2b3e0d18017ecdb618f
e0d2829fe2030032318f1fe803805fea5da0dfdf
50601d6a5380aeb4f060021defd0946ebcd78dfdf
1d72318cfd723ed5b26",VAL "7273"
1030 DATA "fca7ed52200e3e04cd01163e0cd73
e03cd0116c9193a28fc3c3228fc4f3a24fcb9300
3cda9fd46234e232323e5cd2b2def312e38cdf12
b3a22fcc6043225fc91",VAL "6645"
1040 DATA "473e20d710fbcd32de1c38afcf5e
53e04cd0116e1f1fe90302d1112fccd66fd30053
e3fd7182fd68032fff907074f0600eb2139fe0
9eb06081ad71310fb11",VAL "7353"
1050 DATA "1afccd66fd180fd64f1102fccd66f
dd7110afccd66fde53e03cd0116e10601cd78fd2
3c38afcf4f1aa720033779c947131ad71310fb79a
7c93a25fc803225fc4f",VAL "7690"
1060 DATA "3a23fcb9d03e0dd73a22fcc6054f8
03225fc41040528053e20d710fb3a28fc3c3228f
c4f3a24fcb9d0e53a29fca728083e04cd01163e0
cd73e03cd01163a22fc",VAL "6401"
1070 DATA "c605473e20d710fb21b9fe7e3c280
53dd72318f7e13e033228fc3229fcc9030602030
60805030405050404040404040404040404040
4050304050504040405",VAL "3455"
1080 DATA "02020206060406080508060707080
70806080507070809060508080606060906080a0
50506070807060705070607080605060b0505060
70807ff818181818181",VAL "1399"
1090 DATA "ffff818181f1f1f1fffff1f1f1818
181fffff1f1f1f1f1f1fffff8181818f8f8fffff8
18181fffff1f1f1f1f1f18f8f8fffff1f1f1fff
fffff8f8f8f818181",VAL "13170"
1100 DATA "ffff8f8f8ff1f1f1ffffffffff818
181ffffffffff1f1f1fffff8f8f8f8f8f8fffff8
f8f8ffffffffff8f8f8f8ffffffffff8f8f8ffff
fffff4c697374696e67",VAL "13123"
1110 DATA "206f662066696c653a200000000000
0000000000000d0dff",VAL "1064"
```

```
0 REM 16K LOADER
0 REM Gecreeerd met HEXLOAD
30
40 DEF FN F(X$)=16*(CODE X$(1)-48-39*(
X$(1)>"9"))+CODE X$(2)-48-39*(X$(2)>"9")
50 LET ADR=VAL "31744": LET L=1000
60 READ H$: LET SOM=0
70 FOR C=1 TO LEN (H$) STEP 2
80 LET WRD=FN F(H$(C TO C+1))
90 POKE ADR,WRD
100 LET ADR=ADR+1
110 LET SOM=SOM+WRD
120 NEXT C
130 READ TOTAAL
135 IF TOTAAL<>SOM THEN PRINT "FOUTE CH
EKSOM IN REGEL ";L: STOP
140 IF TOTAAL=SOM THEN PRINT "Regel ";L
;" OK"
145 LET L=L+10
150 GO TO 60
1000 DATA "1828031b2d01000000000031b2d000
0000000051b2a0508000000000000000000000
8483e00000000002a535c23232323eeabe200b2
311ca7e010a00edb018",VAL "2158"
1010 DATA "0a21ca7e060a363f2310fb3efa322
87caf3229fc2a0b5c7cb5281ee5dde1dd6e0cdd6
60d23cd6e1922267cdd6e04dd6605cd6e192b3e0
d1812211027cd6e1922",VAL "5792"
1020 DATA "267c2a535c2b3e0d18017ecdb618f
e0d2829fe2030032318f1fe803805fea5da0d7df
50601d6a5380aeb4f060021de7d0946ebcd78dfdf
1d72318cfd723ed5b26",VAL "6761"
```



```

1030 DATA "7ca7ed52200e3e04cd01163e0cd73
e03cd0116c9193a287c3c32287c4f3a247cb9300
3cda97d46234e232323e5cd2b2def312e38cdf12
b3a227cc60432257c91",VAL "5749"
1040 DATA "473e20d710fbcd032de1c38a7cf5e
53e04cd0116e1f1fe90302d11127ccd667d30053
e3fd7182fd68032fff90707074f0600eb21397e0
9eb06081ad71310fb11",VAL "6841"
1050 DATA "1a7ccd667d180fd64f11027ccd667
dd7110a7ccd667de53e03cd0116e10601cd787d2
3c38a7c4f1aa720033779c947131ad71310fb79a
7c93a257c8032257c4f",VAL "6410"
1060 DATA "3a237cb9d03e0dd73a227cc6054f8
032257c41040528053e20d710fb3a287c3c32287
c4f3a247cb9d0e53a297ca728083e04cd01163e0
cd73e03cd01163a227c",VAL "5377"
1070 DATA "c605473e20d710fb21b97e7e3c280
53dd72318f7e13e0332287c32297cc9030602030
60805030405050404040404040404040404040
4050304050504040405",VAL "3071"
1080 DATA "02020206060406080508060707080
70806080507070809060508080606060906080a0
50506070807060705070607080605060b0505060
70807ff8181818181",VAL "1399"
1090 DATA "ffff818181f1f1f1fffff1f1818
181fffff1f1f1f1f1f1ffff8181818f8f8fffff8
18181fffff1f1f1f18f8f8fffff1f1f1fff
fffff8f8f8f818181",VAL "13170"
1100 DATA "ffff8f8f8f1f1f1fffff818
181fffff1f1f1f1fffff8f8f8f8f8f8fffff8
f8f8fffff1f1f1f1fffff8f8f8fffff1f1f1fff
fffff4c697374696e67",VAL "13123"
1110 DATA "206f662066696c653a200000000000
000000000000d0dff",VAL "1064"

```

```

1 REM Ebert Zwart
5 REM Llistinstall
6 DEF FN a(f,1)=USR 64512
10 DIM a(7):CLS
20 PRINT "Installeerprogramma BASIC-li
ster"
30 PRINT "U gaat nu het list programm
a""aanpassen voor uw printer."
32 PRINT "Heeft u een 16K (0) of een
48K""(1) versie: ";
33 LET A$=INKEY$: IF A$="" THEN GO TO
33
34 IF A$="0" THEN LET base=31746: GO T
O 40
35 IF A$="1" THEN LET base=64514: GO T
O 40
36 GO TO 33
40 PRINT ""LADEN lister code": LOAD "
LISTER"CODE base-2,737
45 CLS : PRINT "Wat zijn de control ch
aracters""om UDG te activeren? (max 7):
"
50 GO SUB 500
60 PRINT "Wat zijn de control characte
rs""om UDG te deactiveren?"
70 GO SUB 500
80 PRINT "Wat zijn de control characte
rs""om grafische mode te activeren?"
90 GO SUB 500
100 PRINT "Wat zijn de control characte
rs""om graf. mode te deactiveren?"
110 GO SUB 500
120 PRINT "Wat is uw linker kantlijn? "

```

```

130 INPUT "minimaal 1: ";a: IF a<1 THEN
GO TO 130
140 POKE base,a: PRINT a
145 PRINT "Wat is uw rechter kantlijn?"
";
150 INPUT "minimaal ";(a+31);": ";b: IF
b<a+31 THEN GO TO 150
160 POKE base+1,b: PRINT b
170 PRINT "Wat is uw laatste prinregel"
"per blad? ";
180 INPUT "Minimaal 5: ";c: IF c<5 THEN
GO TO 180
190 POKE base+2,c: PRINT c
200 CLS : PRINT "Lister is geïnstalleerd
. Het ""wordt nu gesaved met: ""SAVE ""
LISTER"" CODE ";base-34;","727"
205 SAVE "LISTER"CODE base-34,727
210 STOP
500 FOR n=0 TO 6
510 INPUT "Alleen ENTER is stoppen: ";
LINE a$
520 IF a$="" THEN GO TO 550
525 PRINT a$;" ";
530 LET a(n+1)=VAL a$
540 NEXT n
550 POKE base,n
560 FOR m=1 TO n
570 POKE base+m,a(m)
580 NEXT m
590 LET base=base+8
595 PRINT ""
600 RETURN

```

UITLEG ASSEMBLER LISTING

R10: Voor 16K gebruikers moet dit #7C00 zijn.

R30-140: Declaratie ROM routine labels en systeemvariabelen.

R180-270: Declaratie programmavariabelen

R290-460: HL wordt met het beginadres van het BASIC programma geladen. Allereerst wordt gekeken of op de 1e regel een REM statement staat. Zo ja, wordt de naam hierachter naar de kopregel verplaatst, zo nee, worden ? in de kopregel gezet.

R520-550: Hier wordt gekeken of deze routine is aangeroepen via een FN statement (DEFADD <> 0).

R560-680: Als dat zo is, worden de parameters opgehaald en de beginadressen van deze regels berekent.

R690-750: Als dat niet zo is, wordt de hoogste regel op 10000 gesteld en de laagste op 0.

R770-840: Gekeken wordt of het current character een 'Enter' is (einde regel). Als dat zo is, wordt naar ENTER gesprongen. Daarna wordt gekeken of het character te printen is (Spatie of groter).

R860-930: De volgende controles worden uitgevoerd: Is het een character kleiner dan 128? Zo nee, is het een grafisch character? Zo nee, is het een keyword?

R940-1000: Als het een keyword is, wordt de lengte van dit keyword opgezocht en in B gezet.

R1010-1050: Als het een gewoon character is, staat in B de waarde 1. Er wordt gecontroleerd of het teken met lengte B nog op de regel past. Vervolgens wordt het character afgedrukt.

R1070-1120: Controle of dit de laatste regel was.

R1130-1190: Print een FormFeed en Return naar BASIC

R1200-1280: Er wordt 1 bij de regelteller opgeteld en gekeken of de overflow regel bereikt is. Zo ja, wordt de kopregel afgedrukt in PR_HDR

R1290-1520: Het volgende regelnummer wordt in BC gezet en op de Calculator stack geplaatst. Nu wordt er een string van gemaakt om te kijken hoelang het regelnummer in characters is. Tevens wordt het nummer op de stack gedupliceerd om later uit te printen met PR_FP.

R1540-1590: Het "b" kanaal wordt geopend

R1600-1610: Kijk of het een UDG is.

R1620-1670: Print de control characters voor grafische mode. Als er geen control characters zijn (Carry Set), wordt een ? afgedrukt.

R1680-1840: Zoek in TABLE1 de te zenden 8 bytes op en zend ze naar de printer.

R1860-1910: Print control characters om UDG te activeren, vervolgens de UDG zelf en de control characters om de UDG te deactiveren.

R1930-2000: Open het "t" kanaal weer en ga terug voor het volgende character.

R2020-2170: Deze subroutine print de control characters geadresseerd met DE. Als het aantal control characters 0 blijkt te zijn, wordt carry 1 anders wordt carry 0.

R2190-2430: De lengte in B wordt opgeteld bij de caracterteller. Als deze niet groter is dan de Rechterkantlijn, wordt er verder niets gedaan. Zo niet, wordt er naar een nieuwe regel gesprongen, spaties geprint naar gelang de linkerkantlijn en de regelteller verhoogd.

R2450-2730: Deze subroutine print de kopregel. Een formfeed wordt geprint (behalve de eerste keer). Vervolgens wordt de kopregel afgedrukt en de regelteller gereset.

R2750-2880: Dit zijn de lengtes van de keywords beginnende bij RND (165).

R2900-3320: Dit zijn de bytes die naar de printer gestuurd moeten worden voor de grafische karakters (8 per character)

R3340-3380: De kopregel.

Ebert Zwart.

ASSEMBLER-LISTING

10	ORG	#FC00	180	D_UDG	DEFS	8
20			190	T_UDG	DEFS	8
30	DEFADD	EQU #5C0B	200	D_GRA	DEFS	8
40	PROG	EQU #5C53	205	T_GRA	DEFS	8
50	OP_CHN	EQU #1601	210	L_MARG	DEFS	1
60	LN_ADR	EQU #196E	220	R_MARG	DEFS	1
70	PR_FP	EQU #2DE3	230	OF_LNE	DEFS	1
80	NUMBER	EQU #18B6	240	COUNT	DEFS	1
90	STK_BC	EQU #2D2B	250	ENDLST	DEFS	2
100	STKFCH	EQU #2BF1	260	LINE	DEFS	1
110	DUPL	EQU #31	270	FLAG	DEFS	1
120	STR_S	EQU #2E	280			
130	END_C	EQU #38	290	MAIN	LD	HL, (PROG)
140	UDG	EQU #5C7B	300		INC	HL
150			310		INC	HL
160	JR	MAIN	320		INC	HL
170			330		INC	HL

340	LD	A, #EA	; REM
350	CP	(HL)	
360	JR	NZ, NONAME	
370	INC	HL	
380	LD	DE, NAME	
390	LD	BC, 10	
400	LDIR		
410	JR	ENDNM	
420	NONAME	LD	HL, NAME
430		LD	B, 10
440		LD	(HL), "?"
450	INC	HL	
460	DJNZ	\$-3	
470			
480	ENDNM	LD	A, 250
490		LD	(LINE), A
500	XOR	A	
510		LD	(FLAG), A
520		LD	HL, (DEFADD)
530		LD	A, H
540	OR	L	
550	JR	Z, NO_FN	
560	PUSH	HL	
570	POP	IX	
580		LD	L, (IX+12)
590		LD	H, (IX+13)
600		INC	HL

610	CALL LN_ADR	1330	INC HL	2020	PR_ST4	LD C,A
620	LD (ENDLST),HL	1340	INC HL	2030	LD A,(DE)	
630	LD L,(IX+4)	1350	PUSH HL	2040	AND A	
640	LD H,(IX+5)	1360	CALL STK_BC	2050	JR NZ,P1	
650	CALL LN_ADR	1370	RST #28	2060	SCF	
660	DEC HL	1380	DEFB DUPL	2070	LD A,C	
670	LD A,#0D	1390	DEFB STR_S	2080	RET	
680	JR START	1400	DEFB END_C	2090	P1 LD B,A	
690	NO_FN LD HL,10000	1410	CALL STKFCH	2100	INC DE	
700	CALL LN_ADR	1420	LD A,(L_MARG)	2110	P2 LD A,(DE)	
710	LD (ENDLST),HL	1430	ADD A,4	2120	RST #10	
720	LD HL,(PROG)	1440	LD (COUNT),A	2130	INC DE	
730	DEC HL	1450	SUB C	2140	DJNZ P2	
740	LD A,#0D	1460	LD B,A	2150	LD A,C	
750	JR START	1470	SPACES LD A," "	2160	AND A	
760		1480	RST #10	2170	RET	
770	NXBYTE LD A,(HL)	1490	DJNZ SPACES	2180		
780	START CALL NUMBER	1500	CALL PR_FP	2190	CNTR LD A,(COUNT)	
790	CP #0D	1510	POP HL	2200	ADD A,B	
800	JR Z,ENTER	1520	JP NXBYTE	2210	LD (COUNT),A	
810	CP " "	1530		2220	LD C,A	
820	JR NC,PRCHAR	1540	GRAPH PUSH AF	2230	LD A,(R_MARG)	
830	INC HL	1550	PUSH HL	2240	CP C	
840	JR NXBYTE	1560	LD A,4	2250	RET NC	
850		1570	CALL OP_CHN	2260	LD A,#0D	
860	PRCHAR CP #80	1580	POP HL	2270	RST #10	
870	JR C,NO_GRA	1590	POP AF	2280	LD A,(L_MARG)	
880	CP #A5	1600	CP #90	2290	ADD A,5	
890	JP C,GRAPH	1610	JR NC,PR_UDG	2300	LD C,A	
900	NO_GRA PUSH AF	1620	LD DE,D_GRA	2310	ADD A,B	
910	LD B,1	1630	CALL PR_ST4	2320	LD (COUNT),A	
920	SUB #A5	1640	JR NC,G1	2330	LD B,C	
930	JR C,NO_TOK	1650	LD A,"?"	2340	INC B	
940	EX DE,HL	1660	RST #10	2350	DEC B	
950	LD C,A	1670	JR ENDGRA	2360	JR Z,P_NOSP	
960	LD B,#00	1680	G1 SUB #80	2370	SPACE LD A," "	
970	LD HL,TABLE	1690	LD (63999),A	2380	RST #10	
980	ADD HL,BC	1700	RLCA	2390	DJNZ SPACE	
990	LD B,(HL)	1710	RLCA	2400	P_NOSP LD A,(LINE)	
1000	EX DE,HL	1720	RLCA	2410	INC A	
1010	NO_TOK CALL CNTR	1730	LD C,A	2420	LD (LINE),A	
1020	POP AF	1740	LD B,0	2421	LD C,A	
1030	RST #10	1750	EX DE,HL	2422	LD A,(OF_LNE)	
1040	INC HL	1760	LD HL,TABLE1	2423	CP C	
1050	JR NXBYTE	1770	ADD HL,BC	2430	RET NC	
1060		1780	EX DE,HL	2440		
1070	ENTER RST #10	1790	LD B,8	2450	PR_HDR PUSH HL	
1080	INC HL	1800	NXT LD A,(DE)	2460	LD A,(FLAG)	
1090	LD DE,(ENDLST)	1810	RST #10	2470	AND A	
1100	AND A	1820	INC DE	2480	JR Z,PR_HD2	
1110	SBC HL,DE	1830	DJNZ NXT	2490	LD A,#04	
1120	JR NZ,NO_END	1831	LD DE,T_GRA	2500	CALL OP_CHN	
1130	LD A,#04	1832	CALL PR_ST4	2510	LD A,#0C	
1140	CALL OP_CHN	1840	JR ENDGRA	2520	RST #10	
1150	LD A,#0C	1850		2530	PR_HD2 LD A,#03	
1160	RST #10	1860	PR_UDG SUB #4F	2540	CALL OP_CHN	
1170	LD A,#03	1870	LD DE,D_UDG	2550	LD A,(L_MARG)	
1180	CALL OP_CHN	1880	CALL PR_ST4	2560	ADD A,5	
1190	RET	1890	RST #10	2570	LD B,A	
1200	NO_END ADD HL,DE	1900	LD DE,T_UDG	2580	E3 LD A," "	
1210	LD A,(LINE)	1910	CALL PR_ST4	2590	RST #10	
1220	INC A	1920		2600	DJNZ E3	
1230	LD (LINE),A	1930	ENDGRA PUSH HL	2610	LD HL,HEADLN	
1240	LD C,A	1940	LD A,3	2620	E1 LD A,(HL)	
1250	LD A,(OF_LNE)	1950	CALL OP_CHN	2630	INC A	
1260	CP C	1960	POP HL	2640	JR Z,E2	
1270	JR NC,NO_FF	1970	LD B,1	2650	DEC A	
1280	CALL PR_HDR	1980	CALL CNTR	2660	RST #10	
1290	NO_FF LD B,(HL)	1990	INC HL	2670	INC HL	
1300	INC HL	2000	JP NXBYTE	2680	JR E1	
1310	LD C,(HL)	2010		2690	E2 POP HL	
1320	INC HL					


```

2700 LD A,3
2710 LD (LINE),A
2720 LD (FLAG),A
2730 RET
2740
2750 TABLE DEFB 3,6,2,3,6,8,5
2760 DEFB 3,4,5,5,4,4,4
2770 DEFB 4,4,4,4,4,3,4
2780 DEFB 4,4,4,4,5,3,4
2790 DEFB 5,5,4,4,4,5,2
2800 DEFB 2,2,6,6,4,6,8
2810 DEFB 5,8,6,7,7,8,7
2820 DEFB 8,6,8,5,7,7,8
2830 DEFB 9,6,5,8,8,6,6
2840 DEFB 6,9,6,8,10,5
2850 DEFB 5,6,7,8,7,6,7
2860 DEFB 5,7,6,7,8,6,5
2870 DEFB 6,11,5,5,6,7
2880 DEFB 8,7
2890
2900 TABLE1 DEFB #FF,#81,#81
2910 DEFB #81,#81,#81
2920 DEFB #81,#FF,#FF
2930 DEFB #81,#81,#81
2940 DEFB #F1,#F1,#F1
2950 DEFB #FF,#FF,#F1
2960 DEFB #F1,#F1,#81
2970 DEFB #81,#81,#FF
2980 DEFB #FF,#F1,#F1
2990 DEFB #F1,#F1,#F1
3000 DEFB #F1,#FF,#FF
3010 DEFB #81,#81,#81
3020 DEFB #8F,#8F,#8F
3030 DEFB #FF,#FF,#81
3040 DEFB #81,#81,#FF
3050 DEFB #FF,#FF,#FF
3060 DEFB #FF,#F1,#F1
3070 DEFB #F1,#8F,#8F
3080 DEFB #8F,#FF,#FF
3090 DEFB #F1,#F1,#F1
3100 DEFB #FF,#FF,#FF
3110 DEFB #FF,#FF,#8F
3120 DEFB #8F,#8F,#81
3130 DEFB #81,#81,#FF
3140 DEFB #FF,#8F,#8F
3150 DEFB #8F,#F1,#F1
3160 DEFB #F1,#FF,#FF
3170 DEFB #FF,#FF,#FF
3180 DEFB #81,#81,#81
3190 DEFB #FF,#FF,#FF
3200 DEFB #FF,#FF,#F1
3210 DEFB #F1,#F1,#FF
3220 DEFB #FF,#8F,#8F
3230 DEFB #8F,#8F,#8F
3240 DEFB #8F,#FF,#FF
3250 DEFB #8F,#8F,#8F
3260 DEFB #FF,#FF,#FF
3270 DEFB #FF,#FF,#FF
3280 DEFB #FF,#FF,#8F
3290 DEFB #8F,#8F,#FF
3300 DEFB #FF,#FF,#FF
3310 DEFB #FF,#FF,#FF
3320 DEFB #FF,#FF
3330
3340 HEADLN DEFM "Listing of "
3350 DEFM "file: "
3360 NAME DEFS 10
3370 DEFB #0D,#0D,#FF
3380
3390
3400 ZZZ

```

UNIVERSELE HEX/DEC CONVERSIE J.C. HOLLANDER

Er zijn meer van deze conversieprogramma's gepubliceerd. Deze heeft het voordeel dat deze zowel kleine- als hoofdletters accepteert en rekent tot maar liefst $9.9 \text{ E}+38$ decimaal en tot 33 HEXA getallen !! Men kan dit koppelen aan evt. eigen programma's. Een eenvoudige GO TO 7000 brengt U in het conversieprogramma.

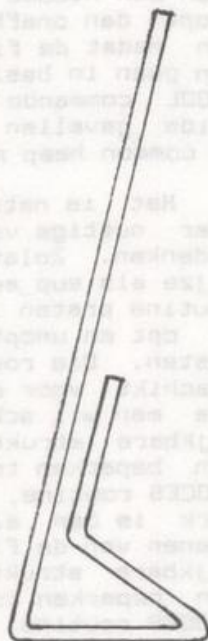
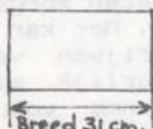
```

7000 GO TO 7200
7001 REM * J.C.Hollander
      * Oude Tonge
7003 REM hex naar dec.
7004 CLS : INPUT "HEX (0 t/m F) : "; Z$: P
      RINT AT 12,1;Z$;" (Hex.)=";
7005 LET Y0=LEN Z$: LET Y1=0
7006 FOR N=1 TO Y0
7008 LET Y2=16^(Y0-N): LET U$=Z$(N): LET
      Y4=CODE U$
7010 IF Y4>=65 AND Y4<=70 THEN LET Y5=Y4
      -55
7015 IF Y4<=57 THEN LET Y5=Y4-48
7017 IF Y4>=97 AND Y4<=102 THEN LET Y5=Y
      4-87
7020 LET Y6=Y5*Y2: LET Y1=Y1+Y6
7021 NEXT N
7025 PRINT " ";Y1;" DEC.": PAUSE 0: GO
      TO 7200
7100 REM **DEC. NAAR HEX.
7110 CLS : INPUT "DEC.GETAL ?:"; Z
7115 PRINT AT 12,3;Z;" DEC.=";
7120 LET X=Z: LET W=1+INT (LN Z/LN 16)
7125 DIM X(W): DIM R(W)
7130 FOR N=W TO 1 STEP -1
7135 GO SUB 7180: NEXT N
7140 PRINT """" = HEX ";
7150 FOR N=W TO 1 STEP -1
7160 GO SUB 7190: NEXT N
7170 PAUSE 0: GO TO 7200
7180 LET X(N)=INT (Z/16^(N-1)): LET R(N)
      =X(N)*16^(N-1)
7185 LET Z=Z-R(N): RETURN
7190 IF X(N)>9 THEN PRINT CHR$ (X(N)+55)
      ;" ";: RETURN
7195 PRINT CHR$ (X(N)+48);" ";: RETURN
7200 REM *** MENU ***
7205 CLS : PRINT " M E N U"
7210 PRINT "[D]= VAN DEC NAAR HEX." "[H]
      = VAN HEX NAAR DEC." "" "WERKT ONGELIMITE
      ERD MET CAPITALS en klein door elkaar to
      t dec. E+38 of...tot 33 HEXA-GETALLEN."
7220 PRINT "" "MAAK UW KEUZE: D of H"
7225 IF INKEY$="h" OR INKEY$="H" THEN GO
      TO 7001
7230 IF INKEY$="d" OR INKEY$="D" THEN GO
      TO 7100
7235 GO TO 7225

```


LIST-TIP

Een handige tip van R. Crèvecoeur uit Epe, om het intypen van listings een stuk gemakkelijker te maken. Haal bij een huishoudwinkel een kookboekenstandaard. Dat is een simpele gebogen plaat perspex in de vorm zoals je op de tekening ziet. Het kost iets van een tien gulden (180 fr), en het vermijdt dat je boeken of tijdschriften nog moet openplooiën, verzwaren of wat dan ook. Je blad blijft netjes rechtop staan, en je typt maar. (Kijk 's bij Blokker...)



QL

FILE COMPACTER

DEEL 2

A. FLOMP

OVER DE SUP_ASM FILE

Zowel cpt als uncpt maken gebruik van de file die sup_asm heet. Deze file bevat de code zorgt dat cpt en uncpt een stel channel_id's toegeworpen krijgen en zich dus niet hoeven bezig te houden met het openen van files en wat daaraan vast zit. Dit heeft twee voordelen, ten eerste is dit openen van files voor veel jobs een ongeveer gelijke bezigheid zodat sup_asm voor veel programma's te gebruiken is. Ten tweede is het handig om deze code apart te kunnen verbeteren, als we sup_asm door iets beters vervangen zijn meteen alle programma's die er gebruik van maken verbeterd.

De huidige versie van sup_asm herkent twee verschillende aanroepen van een job, de "EXEC prog" variant, waarbij een window geopend wordt en om in- en uitvoer files gevraagd wordt, en de "EX prog,file1,file2" variant waarbij prog wordt losgelaten op file1 en file2. De eerste variant heeft zijn window in de rechter bovenhoek, als positie of afmetingen niet bevallen hoeft alleen sup_asm aangepast worden, alle programma's doen het na opnieuw door de assembler gehaald te zijn dan goed.

Uiteraard is het goed mogelijk om sup_asm te vervangen door iets mooiers. De "verbinding" tussen sup_asm en de files die hem gebruiken gaat als volgt. Sup_asm roept steeds de routine PROCES in aan met:

- a0 = channel_id invoer file
- a3 = channel_id uitvoer file
- a4 = pointer naar "data area"
- a7 = stack pointer

Hierbij is de "data area" een stuk geheugen waarin PROCES zijn variabelen kwijt kan. De routine PROCES moet alle registers bewaren behalve d0, waarin een standaard QDOS error code terugkomt, als alles goed is gegaan komt hierin 0 te staan.

Naast deze zeer vaste afspraken zijn er ook nog enige die specifiek zijn voor sup_asm, maar die niet voor elke vervangende routine zo behoeven te zijn. Dit zijn:

- a6 en a5 bevatten bij de aanroep van PROCES begin respectievelijk eind van de job.

- Het programma dat sup_asm gebruikt is een job en de file begint met een standaard job header met als eerste instructie "bra.s start", en voor de naam van de job staat het label "job_name". Dit zorgt ervoor dat bij exec-en van de job de sup_asm code wordt aangeroepen. Het is het handigst om de file sup_asm direct hierna te includen.

- De file die sup_asm gebruikt zorgt zelf voor voldoende "data area", die met een DATA (of soms SIZE) opdracht.

Een verdere afspraak is nog dat eventuele opties in vervangers van sup_asm met behulp van daartoe gespecificeerde labels worden doorgegeven. Zo zou men bijvoorbeeld een optie in kunnen bouwen dat als sup_asm wordt geinclude met label OPT_DELETE = 1, dan wordt de invoer file na gebruik gewist.

Voorbeelden van vervangers.

Een mogelijke vervanging van sup_asm zou een stuk code kunnen zijn dat in plaats van om filenames te vragen het mogelijk maakt deze met cursor toetsen (of een muis) uit een op het beeldscherm weergegeven directory te "plukken".

Een andere mogelijkheid is het maken van een "special" job voor de Toolkit, die een aanroep "EX cpt, file1, file2," uitvoert zoals de huidige versie "EX cpt, file1, file1_cpt", "EX cpt, file2, file2_cpt", zou uitvoeren. Hiermee zou men snel een aantal files tegelijk kunnen verwerken.

Weer een andere mogelijkheid is het verbouwen van cpt en uncpt tot basic commando's. Dit is wat gecompliceerder, de "DATA xxxx" zou uit de files moeten verdwijnen, evenals de jobheaders. Verder zou men PROCES in supervisor mode moeten aanroepen omdat die er geen rekening mee houdt dan a6 in superbasic gebruikt wordt als een base pointer die

tussen de instructies door gewijzigd kan worden. Een andere manier om dit aan te pakken zou zijn om de job header te laten staan en het basic commando gewoon een nieuwe job laten aanmaken. Deze loopt dan onafhankelijk van basic, zodat men nadat de files geopend zijn verder kan gaan in basic, net zoals bij de vele SPOOL commando's die in omloop zijn. In beide gevallen zou men de workspace op de common heap moeten halen.

Het is natuurlijk mogelijk nog veel meer nuttige vervangers voor sup_asm te bedenken. Zolang deze op vergelijkbare wijze als sup_asm dat doet met de PROCES routine praten is het relatief eenvoudig om cpt en uncpt met die routines uit te rusten. Die routines zijn dan ook weer geschikt voor alle volgende programma's die men wil schrijven en die een vergelijkbare structuur hebben. Men kan zich dan beperken tot het schrijven van de PROCES routine, wat natuurlijk minder werk is dan als men ook nog voor het openen van de files zou moeten zorgen. Ijkbare structuur hebben. Men kan zich dan beperken tot het schrijven van de PROCES routine, wat natuurlijk minder werk is dan als men ook nog voor het openen van de files zou moeten zorgen.

ASSEMBLER-LISTING

1. UNCOMPACTER

```
* Uncompacter, huffman-code terug naar ascii.
* version 2.00      (c) A. Plomp 1986
* n.a.v. Byte mei '86.
```

```
bra.s start      Jobs start in support
dc.l 0
dc.w $4afb
job_name dc.w 9    rest van job header
dc.b 'Uncompact',0
```

```
* Include support routine, zorgt voor correcte aanroep proces.
```

```
get 'fdk1_sup_asm'
```

```
* Definitie van enkele QDOS constanten.
```

```
io.pend equ 0
io.fstrg equ 3
io.sstrg equ 7
fs.heads equ $46
```

```
err.ef equ -10
```

```
* Start van werkelijke 'uncompact routine'.
```

```
* Alle registers behalve d0 blijven onveranderd, stack blijft ook
```

```
* onveranderd.
```

```
* INPUTS : a0 = channel id van invoer file
```

```
*          a3 = channel id van uitvoer file
```

```
*          a4 = pointer naar data space
```

```
* OUTPUTS: d0 = qdos error code ( nul = alles OK )
```

```
proces movem.l d1-d7/a0-a6,-(sp) bewaar alle registers
move.l a7,err_stck(a4) bewaar stack pointer voor noodgevallen
move.l a0,inp_id(a4) ze channel id's in variabelen
move.l a3,outp_id(a4)
```



```

*      type0      move.l    hd.len(a4),d0      END IF ;
               tst.l      d0      d0 := lengte file (ge-uncompact!)
               ble.s      end_cpt      IF lengte = 0 THEN
                                   EXIT proces ;

```

```
lea    out_buf(a4),a6    init uitvoer buffer
move.w #out_len,d6       .
```

* d3 & a3 verwijzen voor later gebruik naar de wortel van de boom.

```

    moveq    #0,d3          d3 := 0
    move.l   a4,a3          a3 := [d3]

```

- * Kijk of het een "slechts 1 karakter code" of een "gewoon geval" is.
- * Als het een gewoon geval is moet de boom gelezen worden.

```

        bsr     get_bit      IF get_bit = 0 THEN
        bcc.s   mono_chr     GOTO special geval: slechts 1 cha
*
        bsr.s   rt_entry     lees boom
*
                                END IF

```

* Zet d3 weer op nul. Een move.b naar d3 geeft dan een d3.1 resultaat.

```

moveq    #0,%eax      ; 0 := 0

```

* Nu komt de eigenlijke vertaalslag. (u.tb.151739f,56 d.5voo

```
*
nxt_byte move.l    a4,a3
*
nxt_bit   moveq     #0,d1
          bsr.s      get_bit
          roxl.b      #1,d1
          move.b       left(a3,d1.w),d3
          btst         d1,(a3)
          beq.s        leaf
          move.l        a4,a3
          adda.l        d3,a3
          adda.l        d3,a3
          adda.l        d3,a3
          bra.s         nxt_bit
leaf       move.b       d3,(a6)+
          subq.w        #1,d6
          bgt.s         not_full
          bsr           flush_outp
*
*
not_full  subq.l      #1,d0
          bgt.s        nxt_byte
          bra.s         fl_exit
          leeg write_buffer en END
```


* Het speciale geval dat de boom slechts uit 1 blad bestaat. De file
 * bevatte dan slechts 1 type karakter. Door dit ene karakter 'lengte'
 * maal te herhalen krijgen we de originele file terug.

```
mono_chr bsr.s      rd_byte      d2 := code_in_blad_van_boom ;
*                                     FOR d0 := lengte DOWNT0 1 DO
mc_loop  move.b     d2,(a6)+      WRITE_BYTE( d2 ) ;
      subq.w       #1,d6          IF write_buffer_full THEN
      bgt.s        mc_nxt         leeg de buffer
      bsr          flush_outp    END IF ;
mc_nxt   subq.l     #1,d0          END FOR d0 ;
      bgt.s        mc_loop
```

* We zijn klaar. We hoeven slechts de uitvoer buffer voor het laatst weg
 * te schrijven en terug te keren naar de aanroeper van proces.

```
fl_exit bsr          flush_outp    leeg de buffer
```

* Entry point als zelfs de buffer leeg is.

```
end_cpt  moveq      #0,d0          Geen error
      bra          qd3_ext        Herstel registres & END
```

* Subroutines
 * =====

* Routine om de boom in te lezen. Merk op dat deze aangeroepen moet worden
 * op rt_entry en nergens anders. Er moet gelden d3=0 en a3 = start van array
 * bij die aanroep.
 * INPUTS : d3=0, a3=[0].
 * OUTPUTS: geen. Boom wordt altijd met wortel in array[0] afgeleverd.
 * GEBRUIKT d1,d2 en a1.
 * In d3.w wordt de laatst gebruikte node bijgehouden. De volgende is leeg
 * en dus te gebruiken als we een nieuwe nodig hebben.

* RD_TREE is een routine die een knoop of blad inleest.
 * INPUTS : a1 en d1. De zaak moet in [a1].left als d1=0 en
 * in [a1].right als d1=1.
 * OUTPUTS: geen.
 * GEBRUIKT d2,d3 en a3, invoer buffer gebruikt d4,d5 en a5.

```
rd_tree  bsr.s      get_bit      IF getbit = 0 THEN
      bcs.s        sub_rt        (* blad *)
      bclr         d1,(a1)        Type = blad
      swap         d3             BEWAAR d3.w ;
      bsr.s        rd_byte        lees_byte(d2) ;
      swap         d3             HERSTEL d3.w ;
      move.b       d2,left(a1,d1.w) stop d2 in array.
      rts          ELSE
```

* Kode die een knoop inleest.

```
sub_rt   addq.w     #1,d3          d3 := d3 + 1 ; (* = nieuwe node *)
      addq.l       #3,a3          a3 := [ d3 ] ;
      move.b       d3,left(a1,d1.w) Stop index nieuwe node in array.
      bset         d1,(a1)        Type = knoop
rt_entry move.l     a1,-(sp)        BEWAAR a1 ;
      move.l       a3,a1          lees_boom([d3],links) ;
      moveq        #0,d1          .
      bsr.s        rd_tree        .
      moveq        #1,d1          lees_boom([d3],rechts) ;
      bsr.s        rd_tree        '
      move.l       (sp)+,a1        HERSTEL a1 ;
      rts          END IF ;
```

* Subroutine die een byte inleest. Een byte staat met zijn LSB eerst in de
 * invoer file.
 * INPUTS : geen
 * OUTPUTS: d2.b is ingelezen byte.
 * GEBRUIKT d3, invoer buffer gebruikt d4,d5 en a5.

```
rd_byte  move.w     #7,d3          FOR d3 := 7 DOWNT0 0
```



```
rd_nxt    bsr.s      get_bit      d2.b := (d2.b >> 1)+(128*get_bit) ;
          roxr.b     #1,d2
          dbra       d3,rd_nxt    END FOR d3:
          rts
```

* Belangrijke subroutine die een bit inleest. Het ingelezen bit komt in de
 * X en C flags te staan. Normaal wordt die gebruikt door middel van:
 * - ROXR en ROXL,
 * - BCC en BCS.
 * De invoer file is gebufferd in d4,d5 en a5, die tussen aanroepen van
 * deze subroutine dan ook niet veranderd mogen worden.
 * INPUTS : geen.
 * OUTPUTS: X en C flags.
 * GEBRUIKT d4,d5 en a5.

```
get_bit   dbra       d5,d4_bit      IF bit_buffer_leeg THEN
          swap       d5              d5.w omschakelen naar long_word_buffer
          dbra       d5,buf_bit     IF long_word_buffer_leeg THEN
          movem.l    d0-d3/a0-a1,-(sp) BEWAAR registers ;
          move.w     #inp_len,d2     Lees_buffer(inp_buf,inp_len) ;
          lea        inp_buf(a4),a5
          move.l     a5,a1
          bsr.s      fstrg
          tst.w      d1
          beq.s      qd3_err         IF niets_gelezen THEN
          subq       #1,d1           RAISE ; (* end_of_file *)
          asr.w      #2,d1           END IF ;
          move.w     d1,d5           d5.w := (d1-1) >> 2 ;
          movem.l    (sp)+,d0-d3/a0-a1 HERSTEL registers ;
*
buf_bit   move.l     (a5)+,d4        Haal long_word uit buffer ;
          swap       d5              d5.w omschakelen naar bit_buffer
          move.w     #31,d5          nog 31 bits in bit_buffer
*
d4_bit    roxl.l     #1,d4          END IF ;
          rts                    Haal een bit uit bit_buffer.
```

* Routine die een string bytes van de invoer file leest.
 * INPUTS : a1 = start van buffer,
 * d2 = lengte buffer.
 * OUTPUTS: d1 = aantal ingelezen bytes,
 * a1 = net voorbij laatst ingelezen byte.
 * d0 = error kode (0 of end_of_file)
 * GEBRUIKT d3 en a0.

```
fstrg     move.l     inp_id(a4),a0    QD3 met channel_id van invoer file
          moveq      #io.fstrg,d0     en d0 = io.fstrg.
```

* Routine die een TRAP #3 doet met time-out oneindig en als enig mogelijke
 * error in d0 = end of file.
 * INPUTS : als TRAP #3 behalve d3.
 * OUTPUTS: als TRAP #3 behalve d0, die is altijd 0 of err.ef.

```
qd3       moveq      #-1,d3          time-out := oneindig ;
          trap       #3              TRAP #3 ;
          tst.b      d0              IF error
          beq.s      qd3_rts         AND error <> end of file THEN
          cmpi.b     #err.ef,d0      RAISE
          beq.s      qd3_rts         END IF ;
```

* Routine die een error dwars door subroutine aanroepen direct teruggeeft
 * aan de aanroeper van proces.
 * INPUTS : d0 = error kode.
 * qd3_ext entry point voor als de stack in orde is.

```
qd3_err   move.l     err_stck(a4),a7  Gebruik error_stackpointer
qd3_ext   movem.l    (sp)+,d1-d7/a0-a6 Herstel alle registers behalve d0
qd3_rts   rts
```

* Routine die de uitvoer buffer leegt. De uitvoer buffer gebruikt d6 en a6.
 * INPUTS : d6 bevat aantal ongebruikte bytes in uitvoer buffer.
 * OUTPUTS: a6 = start van buffer,
 * d6 = lengte buffer. (* dus buffer geherinitialiseert *)


```

flush_outp
    lea        outp_buf(a4),a6    a6 := start van buffer ;
    movem.l    d0-d3/a0-a1,-(sp)  BEWAAR registers ;
    move.w     #outp_len,d2       d2.w := buffer_lengte-ongebruikte_bytes ;
    sub.w      d6,d2
    beq.s      empty             IF d2.w <> 0 THEN
    move.l      outp_id(a4),a0     schrijf_buffer_weg
    move.l      a6,a1
    moveq       #io.sstrg,d0
    bsr.s      qd3                END IF ;
empty        movem.l    (sp)+,d0-d3/a0-a1  HERSTEL registers ;
    move.w     #outp_len,d6       d6.w := buffer_lengte.
    rts

```

* EINDE subroutines.

* DATA TYPEN:

* =====

* record def. Lengte_record = 3 bytes.

* Eerste byte is de "tag". Bit 0 hiervan dient voor het "left" veld en
 * bit 1 voor het "right" veld. Een 1 betekend een knoop, een 0 een blad.

```

left        equ        1          2nd byte
right       equ        2          3th byte

```

* Array[0..255] of record, begint direct op (a4).

```
array_size   equ        3*256

```

* Gebruik array:

* In het het array wordt een boom structuur op geslagen. Elk record in
 * het array is een node. Een node heeft twee subbomen (linker en rechter).
 * Een subboom kan een knoop of een blad zijn. Als het een knoop is bevat
 * het "left" (of "right") veld van de "parent node" de index van de node
 * die de wortel is van de subboom en het bijbehorende bit in het "tag"
 * veld een "1". Als het een blad is dan bevat het "left" (of "right") veld
 * van de "parent node" de karakter kode van dat blad en het bijbehorende
 * bit in het tag veld een "0".
 * De wortel van de boom zit in het eerste record van het array. Omdat ook
 * die altijd twee subbomen moet hebben is het niet mogelijk zo een boom
 * op te slaan die slechts uit 1 blad bestaat.
 * De boom wordt gebruikt om de huffman code te decoderen. Als er een 0
 * binnen komt neemt men de linker subboom van de plaats in het array waar
 * men is. Bij een 1 neemt men de rechter subboom. Is de genomen subboom
 * een blad dan hebben we een byte en gaan we terug naar de wortel van de
 * boom.

* i/o buffers, eerst komt na array input buffer, dan output buffer

```

inp_len      equ        1024      1k buffer ( op de gok )
outp_len     equ        1024      1k buffer ( op de gok )
inp_buf      equ        array_size
outp_buf     equ        inp_buf+inp_len

```

* Gebruik buffers:

* Invoer buffer: Deze buffer gebruik d4, d5 en d6. In d4 worden steeds 32
 * bits gebufferd. Het aantal bits dat nog in d4 zit staat in d5.w. De
 * ander helft van d5 bevat het aantal longwords dat nog in het geheugen
 * zit. a5 wijst naar het volgende long word in die buffer. Het algoritme
 * gaat als volgt:
 * - Als d4 geen bits meer bevat dan
 * - Als ook in het geheugen geen long words meer zitten dan
 * - Lees een nieuwe lading bytes uit de file
 * - Haal een nieuw long word voor d4 uit het geheugen
 * - Haal het volgende bit uit d4.
 * Merk op dat d4 en het buffer geheugen na gebruik van dit algoritme
 * nooit helemaal vol zitten. Direct nadat ze geladen zijn wordt er immers
 * weer wat uitgehaald. Helemaal leeg kan wel!
 * Alleen de get_bit routine gebruikt de buffer.
 * Initialiseren gaat met d5 := 0. (noch d4, noch het geheugen bevatten iets)


```
* Uitvoer buffer:
* De uitvoer buffer gebruikt d6 en a6. D6 houdt bij hoeveel plaats er nog
* in het buffer geheugen is. A6 wijst naar de eerste vrij plaats. Het
* algoritme gaat als volgt:
* - Zet de byte in de buffer
* - Als de buffer daardoor volraakt dan wordt hij naar de uitvoer gestuurd.
* Merk op dat de buffer weer nooit helemaal vol zit. Zodra hij vol is wordt
* hij onmiddellijk weer geleegd.
* Om de snelheid op te voeren wordt het plaatsen in de buffer en de
* controle of deze daardoor volraakt steeds direct uitgevoerd op de plaats
* waar het programma een byte kwijt moet kunnen. Er is wel een aparte
* routine voor het wegsturen van de buffer als deze vol is of als het
* programma klaar is.
* Initialiseren gaat als volgt: a6 := eerste adres buffer,
*                                     d6 := lengte buffer
```

```
* Variabelen:
* =====
```

```
inp_id      equ      outp_buf+outp_len (long) chan_id input file
outp_id      equ      inp_id+4        (long) chan_id output file

err_stck     equ      outp_id+4        (long) error_stack pointer

* Buffer voor de header van de file.

header       equ      err_stck+4      header, totaal 14 bytes.

* velden binnen header:
hd.len       equ      header          (long) lengte van de file
hd.access    equ      header+4        (byte) access byte, wat dat ook is
hd.type      equ      header+5        (byte) type byte, 0=data, 1=exec, 2=reloc
hd.xtra      equ      header+6        (8 bytes) extra info, type afh.
```

```
* GEHEUGEN GEBRUIK:
* =====
```

* array	768	256 * 3 bytes
* input buffer	1024	1k buffer
* output buffer	1024	1k buffer
* variabelen	12	3 long words
* header	14	Gdos header = 14 bytes
* stack	512	ruim genoeg
* support	0	support gebruikt zelfde werk geheugen
* -----	+	
* TOTAL	3354	
SIZE	3354	sommige assemblers willen DATA 3354.
end		

2. SUPPORTER

```
* Simpele support routine
* Version 2.00 (c) A. Plomp 1986
```

```
* Zorgt ervoor dat de aanroep 'ex prog,file1,file2' verwerkt wordt tot
* een aanroep van de subroutine proces met:
* INPUT : a0 = channel_id file1, geopend als OPEN_IN, dus share.
*         a3 = channel_id file2, geopend als OPEN_OVER, dus overwrite.
*         a4 = eerste adres in vrije data space.
*         a5 = laatste adres van job.
*         a6 = eerste adres van job.
* OUTPUT : d0 = error kode.
```



```
* Een aanroep van de vorm 'exec prog' wordt in een zelfde aanroep naar
* de proces subroutine omgezet; er wordt dan een window geopend en om
* twee filenamen gevraagd. Dit wordt herhaalt totdat er een lege filenaam
* als invoer file wordt opgegeven.
* DATA SPACE : 102 bytes voor een invoer buffer om filenamen in te kunnen
* lezen. Deze buffer hoeft niet bewaard te blijven als
* de proces subroutine wordt aangeroepen. Het enige waarmee
* proces rekening hoeft te houden is dat de eerste 102 bytes
* op (a4) tussen twee aanroepen veranderen.
* Bij gebruik van de vorm 'ex prog,file1,file2 wordt er geen
* geheugenruimte gebruikt.
```

```
* De job header moet beginnen met een sprong naar 'start'.
```

```
* Enkele qdos constanten
```

```
mt.frjob equ      5          Trap #1 code om een job te "force remove"-en
ut.err0  equ      $ca        schrijf error naar #0 van basic
err.bp   equ      -15        bad parameter (error_code)

start     adda.l    a6,a4      a4 := start of dataspace
          adda.l    a6,a5      a5 := end of job
          move.w    (sp)+,d0    d0 := aantal channels van ex
          tst.w     d0         IF d0 = 0 THEN
          beq.s     interact    GOTO interactief
          subq.w    #2,d0      IF d0 <> 2 THEN RAISE bad_parameter
          bne.s     err_bp      .
          move.l    (sp)+,a0    a0 := input channel id
          move.l    (sp)+,a3    a3 := output channel id
          bsr       proces      PROCES
harakiri  move.w    ut.err0,a2  REPORT #0,d0
          jsr       (a2)        .
          move.l    d0,d3      FORCE_REMOVE_JOB( error <= d0,
          moveq     #-1,d1      job <= mezelf )
          moveq     #mt.frjob,d0
          trap      #1          .
```

```
* Produceer een error 'bad parameter'
```

```
err_bp   moveq     #err.bp,d0
          bra.s     harakiri    pleeg zelfmoord
```

```
* Routines om interactief te werken. Er wordt gevraagd om een input en
* een output file.
```

```
* Nog enkele QDOS constanten.
```

```
io.open  equ      1          Trap #2 code om file te openen
io.close equ      2          Trap #2 code om file te sluiten

io.fline equ      2          Trap #3 code om een regel in te lezen

share    equ      1          Bij io.open een file voor input openen
new_excl equ      2          Bij io.open een file voor output openen

ut.con   equ      $c6        Open een CON_ device
ut.scr   equ      $c8        Open een SCR_ device
ut.err   equ      $cc        Schrijf een QDOS error_text naar een channel
ut.mtext equ      $d0        Schrijf een string naar een channel
```

```
* Start van code die de proces routine interactief gebruikt. Er wordt om
* een input en om een output file gevraagd. Dan wordt proces op de de
* input losgelaten waarbij het resultaat in de output file komt.
```

```
* Als eerste wordt de naam van de job in een klein windowtje boven het
* echte 'werk window' gezet.
```

```
interact lea       tag_scr,a1  Open een screen volgens tag_scr
          move.w    ut.scr,a2  .
          bsr.s     vector      .
          lea       job_name,a1 Schrijf de job naam daarin
          bsr.s     mtext       .
          bsr       close_a0    Close dit hulp windowtje weer
```


* Nu wordt het werk window geopend. Voor het grootste deel van de code
* blijft zijn channel id in a0.

```

lea      con_def,a1      Open con_ volgens con_def
move.w   ut.con,a2      .
bsr.s    vector          .

```

* We krijgen nu de main_loop. Er wordt net zolang doorgedaan met het
* vragen om files tot er een lege filenaam wordt gegeven als input file.
* Als er een lege filenaam wordt gegeven als output file dan wordt de zaak
* voortijdig afgebroken: er wordt een nieuwe input filenaam gevraagd.
* N.B. een lege filenaam wordt gegeven door gewoon ENTER in te drukken
* zonder verder iets ingetikt te hebben.

```

* REPEAT main_loop
repeat    lea      qu_inp,a1      WRITE('Input file : ');
          bsr.s    mtext          .
          moveq    #share,d7      get_file(share); (share = open voor input)
          bsr.s    get_file        .
          beq.s    harakiri        IF filenaam was leeg THEN zelfmoord; (d0=0)
          move.l    a3,d6          Bewaar input_channel_id in d6

          lea      qu_out,a1      WRITE('Output file : ');
          bsr.s    mtext          .
          moveq    #new_excl,d7   get_file(new); (new = maak nieuwe file)
          bsr.s    get_file        .
          beq.s    empty_out      IF filenaam niet leeg THEN
          exg      d6,a0          a0 := input_id; d6 := console_id;
          bsr      proces        PROCES
          tst.b    d0            IF error THEN
          beq.s    proces_ok      a0 := console_id;
          exg      d6,a0          d6 := input_id;
          move.w   ut.err,a2      REPORT;
          jsr      (a2)          .

          exg      d6,a0          a0 := input_id, d6 := console_id;
          END IF ;

* proces_ok
          bsr.s    close_a0        Close input file
          move.l    a3,a0          Close output file
close_2nd
          bsr.s    close_a0        .
          exg      d6,a0          a0 := console_id;
          bra.s    repeat        ELSE

empty_out
          exg      d6,a0          a0 := input_id; d6 := console_id;
          bra.s    close_2nd      close input file; a0 := console_id;

* END REPEAT main_loop ;

```

* Subroutine die een string naar een channel schrijft.
* INPUTS : a0 = channel id
* a1 = wijst naar de string.
* OUTPUTS: d0 = error kode.
* GEBRUIKT d1,d2,d3,a2 en a3. a0 blijft hetzelfde.

```

mtext    move.w   ut.mtext,a2      exit via vector

```

* Vector subroutine. Doet een jsr (a2) en controleert op errors.

```

vector    jsr      (a2)
          tst.b    d0
          bne.s    harakiri
          rts

```

* Routine die een filenaam inleest en, als deze niet leeg is, de file
* probeert te openen. Lukt dat niet dan wordt een nieuwe file ingelezen.
* Als de filenaam leeg is wordt dat aan de aanroeper doorgegeven.
* INPUTS : d7 = in/out/share/new etc.
* a0 = con_id


```

* OUTPUTS: a3 = channel id als ZERO_FLAG = FALSE
*           ZERO_FLAG : (TRUE = lege filenaam ,FALSE = filenaam in a3)

*
* REPEAT getfile_loop
get_file moveq    #buff_len,d2    READLN( filenaam$ );
        moveq    #-1,d3
        move.l    a4,a1
        addq.l    #2,a1
        moveq    #io.fline,d0
        trap      #3
        tst.b     d0
        bne       harakiri
        subq.w    #1,d1
        beq.s     exit_gf
        move.w     d1,(a4)
        move.l     a0,a3
        moveq     #-1,d1
        move.l     d7,d3
        move.l     a4,a0
        moveq     #io.open,d0
        trap      #2
        exg       a0,a3
        tst.b     d0
        beq.s     open_ok
        move.w     ut.err,a2
        jsr       (a2)
        bra.s     get_file
        ELSE
        IF error THEN
        RAISE ;
        IF filenaam$ = "" THEN
        RETURN ; (* zero = true, d0 = 0 *)
        (* maak er een qdos string van *)
        a3 := console id;
        a0 := OPEN( job => myself,
                    key => d7,
                    name => filenaam$)
        a0 := console id; a3 := file chan_id;
        IF error bij openen THEN
        REPORT error op console
        ELSE
open_ok moveq     #1,d1
        ZERO_FLAG := FALSE; RETURN
*           END IF;
exit_gf rts
        END REPEAT getfile_loop ;

* Routine om een channel te closen.
* INPUT : a0 = channel id.

close_a0 moveq    #io.close,d0    CLOSE a0
        trap      #2
        tst.b     d0
        bne       harakiri
        RAISE;
        rts

* OPMERKING betreffende de tag_scr, con_def, qu_inp, qu_out en job_name:
* De Metacomco assembler accepteert instructies als "LEA qu_inp,a1". De
* assembler weet dan dat qu_inp programmateller relatief is en niet
* absoluut. Sommige assemblers willen dat er "LEA qu_inp(PC),a1" staat,
* dus dat expliciet wordt vermeld dat het hier een PC relatieve adres is.

* Definities van de verschillende window's: een scr_ om de naam van de
* job in te schrijven en een con_ voor het eigenlijke werk.

tag_scr  dc.b      7,0,7,0        BORDER(0,7): PAPER 7: INK 0
        dc.w      200,11,300,20   scr_200x11a300x20

con_def  dc.b      7,1,0,7        BORDER(1,7): PAPER 0: INK 7
        dc.w      200,82,300,35   con_200x82a300x35

* Twee text strings.

qu_inp   dc.w      14             string "Input file : "
        dc.b      'Input file : '

qu_out   dc.w      14             string "Output file : "
        dc.b      'Output file : '

* De input buffer zit direct op (a4). Dit kan problemen geven aangezien
* PROCES dit ook als werkgeheugen zal gebruiken. Het kan voor SUPPORT
* echter geen kwaad dat de inhoud van de buffer verloren gaat bij een
* aanroep van PROCES. Deze op z'n beurt wordt geacht geen gebruik te
* maken van wat er in het werkgeheugen staat.

buff_len equ      100             100 bytes input buffer

```


one-liner competitie

Speciaal voor de SPECTRUM gebruikers is Deze ONE-LINER COMPETITIE opgezet. Zoals de doorgewinterde spectrumgebruiker weet is de spectrum een van de weinige computers die programmaregels van 'onbeperkte lengte' accepteren.

Het moet voor de spectrumgebruiker een 'koud kunstje' zijn om een programma te schrijven dat uit slechts 1 programmaregel bestaat. Daar wij van de redactie wel eens willen weten waartoe onze lezers in staat zijn, zijn we met deze ONE-LINER COMPETITIE gestart.

Voor de beste one-liner van elk nummer stellen wij een boekenbon van fl 25 beschikbaar en voor de diegene die als tweede eindigt een boekenbon van fl 15. De competitie eindigt in nummer 8 van de SINCLAIR GIDS. Dan wordt de eindwinnaar van deze competitie bekend gemaakt.

De HOOFDPRIJS van deze competitie is een GE TX1000 NLQ PRINTER.

REGLEMENT

Uiteraard zijn er voor deze competitie een aantal regels:

1. Alle lezers van de SINCLAIR GIDS mogen meedoen.
2. De ONE-LINER moet in SINCLAIR BASIC geschreven zijn. (dus geen extended BASIC gebruiken) (Machinetaal wel toegestaan).
3. Om in aanmerking te komen voor publicatie moet het programma binnen zijn voor sluitingsdatum van dat nummer (zie tabel). Alles wat daarna binnenkomt schuift door naar het volgende nummer.
4. Het programma(atje) moet op cassette worden ingestuurd en natuurlijk met uitleg hoe met het programma te kunnen werken.
5. Alle binnengekomen programma's dingen mee naar een der prijzen. (er zijn ook nog troostprijzen in de vorm van spelprogramma's en boeken).
6. Als de ruimte het toelaat zullen wij de ONE-LINERS allemaal publiceren in de SINCLAIR-GIDS.
7. Alle ingezonden programma's dienen eigen werk te zijn. De deelnemers vrijwaren de SINCLAIR GIDS voor eventuele aanspraken van derden op het copyright.



UITSLAG 1e RONDE

We kregen de afgelopen weken een tiental inzendingen binnen voor de eerste ronde van de One-Liner Competitie.

Helaas waren niet alle inzendingen, geschikt voor publicatie. Enkele hadden erbij vermeld dat het een grapje was. Nu de bekroonde inzendingen.

De eerste prijs, een boekenbon van fl 25 gaat naar M.Hellema die het presteerde om een complete UDG-ontwerper in 1 programmaregel onder te brengen. Unaniem was de redactie van mening dat dit programma de eerste prijs toekwam.

De tweede prijs gaat naar P.Rekveld voor een programma zonder naam, die we voor 't gemak tekstmanipulator hebben genoemd. Met dit programma kunt U teksten in het geheugen zetten en deze via een RANDOMIZE USR-opdracht direct oproepen. Er zijn voor dit programma vele gebruiksmogelijkheden, bijv. een computerkrant. Een voordeel is dat een hele pagina tekst nu niet als SCREEN\$ moet worden geSAVED. Een hele geheugenbesparing. Drie troostprijzen gaan naar Bastiaan de Jong uit Nieuw-Vennep, P. Zelderloo uit Asse (België) en M. Jorritsma uit Dongjum (Friesland) voor twee spelletjes en een utility.

(E.R.)

```
Ø>REM Bastiaan de Jong
1Ø PAPER Ø: BORDER Ø: INK 6: C
LS : FOR q=Ø TO 29: READ a: POKE
3ØØØØ+q,a: NEXT q: LET d=15: FO
R a=Ø TO 1ØØØ: LET l=USR 3ØØØØ:
IF SCREEN$ (1Ø,d)<>"Ø" THEN PRIN
T AT 1Ø,d;"V": PRINT AT 2Ø,d+RND
*5-RND*1Ø;"ØØ": PRINT : POKE 236
92,-1: PRINT : LET d=d+(INKEY$=
Ø)-(INKEY$="1"): PRINT AT Ø,Ø;"
Score: ";a;"      mad by Holysoft"
: NEXT a: DATA Ø,6,1,197,33,7,Ø,
17,Ø,Ø,229,2Ø5,181,3,225,17,2Ø,Ø
,167,237,9Ø,125,254,255,32,237,1
93,16,23Ø,2Ø1
```

```
Ø REM P. Rekveld
1 CLS : RESTORE : INPUT "TEKS
T: "; LINE a$: INPUT "Start vana
f geheugenadres: ";g: INPUT "Lij
n: ";l,"Tab: ";t: INPUT "Paperkl
eur (Ø-7): ";paper: INPUT "Inkkl
eur (Ø-7): ";ink: LET n1=175: IN
PUT "Flash (1/Ø) ";f: CLS : RAND
OMIZE (g+13): LET g1=PEEK 2367Ø:
LET g2=PEEK 23671: LET a$=CHR$
22+CHR$ 1+CHR$ t+CHR$ 18+CHR$ f+
CHR$ 16+CHR$ ink+CHR$ 17+CHR$ pa
per+a$: LET z=LEN a$: RANDOMIZE
z: LET z1=PEEK 2367Ø: LET z2=PEE
K 23671: FOR x=g TO g+12: READ m
: POKE x,m: NEXT x: FOR x=g+13 T
O g+12+LEN a$: POKE x,CODE a$(x-
(g+12)): NEXT x: RANDOMIZE USR g
: PAUSE Ø: CLS : LET lengte=z+13
: PRINT "Startadres: ";g;" Len
gte: ";lengte;" "; "Starten via RA
NDOMIZE USR ";g: INPUT "Filenaam
: "; LINE s$: PRINT "Saving ";s
$: SAVE "m";1;s$CODE g,lengte:
PRINT "Verifying ";s$;: VERIFY
"m";1;s$CODE : PRINT " "; INVER
SE 1;"O.K.!:": PRINT "Filenaam:
";s$;" CODE ";g;" ";lengte: DAT
A 2Ø5,n1,13,17,g1,g2,1,z1,z2,2Ø5
,6Ø,32,2Ø1
```



```

1 REM P. Zelderloo
100 INK @: BORDER @: PAPER 7: B
RIGHT 1: CLS : PRINT AT @,@;"Hoe
groot (@/40) ?" : INPUT AT 22-1
,@;AT @,1: LINE j$: POKE 23659,2
: PAUSE 50: PRINT AT @,@;"hoevee
1 vergroten (@/100) ?" : INPUT AT
22-1,@;AT @,1: LINE i$: POKE 23
659,2: PRINT AT @,27:VAL j$: FOR
m=1 TO (VAL j$): PLOT 100+m,10+
m: NEXT m: FOR m=@ TO (VAL j$)-1
: FOR n=1 TO (VAL i$): INK (RND
*6): BEEP @,1,n-35: PLOT m+100,1
8+m+n: NEXT n: NEXT m: PRINT AT
21,10;"DRUK EEN TOETS": PAUSE @:
RUN

```

NR 1



```

1 REM M. Hellema
10 BORDER @: INK 7: PAPER @: B
RIGHT @: CLS : POKE 23618,10: PO
KE 23619,@: FOR n=@ TO 64 STEP 8
: PLOT @,n: DRAW 64,@: PLOT n,@:
DRAW @,64: NEXT n: FOR n=@ TO 6
: FOR m=1 TO 3: PRINT AT n+14,4*
m+12:CHR$ (64+3*n+m): "=" :CHR$ (1
43+3*n+m): NEXT m: NEXT n: PRINT
AT 21,16;"Current=": LET x=@:
LET y=@: PAUSE @: LET i$=INKEY$:
POKE 23620,-(NOT (i$=CHR$ 13)):
POKE 23200+x-32*y,7: LET x=x-(i
$="5")+(i$="8"): LET y=y-(i$="6"
)+(i$="7"): POKE 23200+x-32*y,19
9: POKE 23620,255*(i$="0" OR i$>
="A")+22*(i$="5" AND i$<="8"):
POKE 23620,255*(i$="0")+(i$<>"0"
)*34: PLOT OVER 1;200+x,y: PRINT
AT 21-y,x: OVER 1:CHR$ 143: PO
KE 23620,22: FOR n=@ TO 7: POKE
USR i$+n,PEEK (20665+256*n): NEX
T n: POKE 23620,1

```

```

1 REM M. Jorritsma
2 PAUSE 100: PRINT ?? "druk ee
n toets": PAUSE @: LET t=@: LET
k=10: BORDER 1: PAPER 1: INK 7:
CLS : LET p=21: LET q=15: FOR n=
@ TO 20: PRINT AT n,RND*10: ". .
. . . . .": NEXT n: LET
T y=@: LET x=RND*22: FOR n=@ TO
21: LET x=x+(RND*2): PRINT AT n,
x: INK 4: ": LET c=1 AND INKEY$="
8" AND q<29: LET q=q+c: LET d=1
AND INKEY$="5" AND q>-29: LET q=
q-d: PRINT AT p,q: INK 6: " ? " :
BEEP @.007,30 AND ATTR (n,x)=14:
LET e=1 AND ATTR (n,x)=14: LET
t=t+e: PRINT AT n,x: " ": LET x=x
-(RND*2): PRINT AT @,23;"score
": t: NEXT n: LET k=k-1: PRINT AT
10,10: INK 2;"EINDE" AND k=@: P
OKE 23618,1: POKE 23619,@: POKE
23620,15 AND k<>0

```

FILOSOFT SERIEUS IN SOFTWARE ZX-SPECTRUM

HISOFT BASIC - Een floating-point (!) Basic-compiler voor de Spectrum 48K, Plus, 128, en Spectrum +2. De nieuwste, de snelste en de beste. Maakt van uw Basic-programma's machinecode, die tot 80 keer sneller wordt uitgevoerd. HISOFT BASIC kan bijna alle Spectrum commando's verwerken. Geschikt voor gebruik met cass.rec. microdrive en Opus DD. * cass f 69,00

PROCEED INTERFACE - Is een handig interface dat de Commodore 1541 diskdrive (ja, die van de C'64) in een klap geschikt maakt om als diskdrive voor de Spectrum 48K te fungeren. Bovendien is de PROCEED een parallel (dus voor 'gewone' printers met Centronics aansluiting) printerinterface met meer mogelijkheden dan alle andere (bijv. COPY met zoom, beelduitsnede, willekeurige vergroting). De PROCEED is een aanrader voor iedereen die NIET met Interface I werkt. Duitse handleiding. Van f 290,00 nu voor: f 179,00

TAS-DIARY - Deze elektronische agenda houdt uw afspraken bij met alle voordelen die met computerverwerking van gegevens zijn verbonden. U kunt er ook een dagboek in bijhouden. Alleen voor microdrive, geleverd op: * cass f 45,00

TASWORD DRIE - DE DEFINITIEVE TEKSTVERWERKER voor de Spectrum in combinatie met een of meer microdrives of de Opus Discovery. De geheel Nederl. versie heeft een duidelijke handleiding van 69 pagina's op A4-formaat, en is inclusief de servicekaart (zie de Sinclair Gids no. 1 voor uitgebreide bespreking). Voor Opus Discovery op 3,5" disc, voor microdrive op cart: f 89,00 Inruilers van de Ned. uitgave Tasword Twee betalen zelfs f 25,00 minder voor Tasword Drie indien zij de (originele!) handl. en cass. meezenden!

TASWORD TWEE - Nog steeds DE tekstverwerker voor wie niet met een micro- of disc-drive werkt. Geeft 64 karakters per regel ook op het scherm. cass f 59,00

BETA BASIC 3.0. - Laatste versie van deze veel geprezen Basic uitbreiding. Met WINDOW's PROCedures en CSIZE voor keuze 1 tot 64 of meer tekens per regel. Dit programma maakt de Spectrum Basic tot een van de krachtigste ter wereld. MD-compatible, met uitgebreide Ned. handleiding en demo-programma. cass f 77,50

VERKEER - Test uw kennis van de Nederlandse verkeersborden. 2 delen op een cassette. Het overzicht van de score en de door de gebruiker zelf in te stellen snelheid maken VERKEER geschikt voor scholen, voor rijbewijshouders, als gezelschapsspel, en als test voor mensen met 'jarenlange rijervaring'. f 49,50

TASCOPY - Geeft een afdruk van het beeldscherm op een 'grote' printer. Grootte van de afdruk: A4-formaat (met grijsinten) of zelfs Poster-size. Geschikt voor een reeks van printers (niet alleen de Epson-compatibles, informeer of de uwe er bij zit!) die door Interface I worden aangestuurd. Erg mooi. * cass f 45,00

TASPRINT - Geeft extra (grote en bijzondere) letters aan bijna iedere dotmatrixprinter (ook: Smith Corona Fastext 80). Is zelfstandig, maar ook in combinatie met Tasword te gebruiken. * cass f 45,00

TASDWARF - Werkt samen met Tasword 2. Drukt de tekst dwars op het papier van de Seikosha GP-50S, de ZX-printer of de Alphacom. Het produceert telkens 3 stroken met letters van normale grootte die samen een vel A4 vormen. f 34,50

SPELLINGSBAK 1 EN 2 - Uitgebreide behandeling van de spelling van de Nederlandse taal: de theorie en honderden oefeningen. N.b.: geen spel, maar een educatief programma dat zijn sporen verdient heeft. Voor kinderen en volwassenen van klas 5 van het basisonderwijs. Prijs per deel: f 49,50. Nu samen: f 89,00

REDEKUNDIG ONTLEDEN - Educatief programma met als onderwerp de Nederlandse taal. De theorie en 140 oefeningen. Een behoorlijke uitdaging. cass f 49,50

HISOFT DEVPAC - Voor ieder die werkt met machinecode, of daar mee wil gaan werken: disassembler, assembler en monitor ineen. Een professioneel instrument: 'Masterfile' en 'Tasword' zijn er mee gemaakt! * f 59,00

HISOFT PASCAL - De computertaal Pascal voor de ZX-Spectrum. De compiler genereert code die ook zonder dat de compiler is ingeladen gerund kan worden. Goed! Uitgebreide handleiding. * f 95,00

HISOFT C - De computertaal C is de taal van de toekomst. Ook op zeer grote systemen wordt met deze koning van de hogere programmeertalen gewerkt. Programma's in C geschreven, werken onafhankelijk van de compiler, en zijn met geen of enkel minime wijzigingen op andere computers te gebruiken. cass f 95,00

TOTO - Houdt de Nederlandse voetbalcompetitie bij. cass f 49,50

LETTERBAK - 51 of 64 letters per regel in eigen programma's. cass f 34,50

OMNICALC II - Spreadsheet voor alle berekeningen. Ook grote printers. * f 69,00

OPUS DISCOVERY - VERZAMELDISK - Voor meer dan f 300,00 aan software op een disk: MASTERFILE vs. 9 (de laatste), MPRINT, OMNICALC 2, TASWORD 2 en TASMERGE, met voorbeelden en NEDERLANDSE handleiding voor 1/3 v.d. prijs: disk f 99,00
TASCOPY OPUS - * disk f 69,00
TASPRINT OPUS - * disk f 59,00
BETA BASIC 3.0-OPUS - cass f 77,50
HISOFT BASIC - * cass f 69,00

SPECTRUM 128 - TASWORD 128 - Vergelijkbaar met Tasword Drie, maar met meer ruimte voor tekstopslag (64K). * f 69,00
EN BIJNA ALLE SPECTRUM-48K SOFTWARE
QL
TASCOPY QL - * cart f 55,00
TASPRINT QL - * cart f 79,00

INFORMATIE
Onze databank FILOTEL (050-145 174) is 24 uur per dag bereikbaar voor iedereen, die apparatuur heeft om met Viditel te bellen. Abonnement niet nodig! Service en advies geven we graag en gratis (!) op vrijdag, de servicedag. Voor programma's gemerkt met een ster (*) is enige kennis van de Engelse taal vereist. Andere uitgaven zijn in het Nederlands. Bestellen v.e. programma (o.v.v. uw computermerk) door storting van het verschuldigde bedrag + f 3,50 verz.k. op giro 20792 t.n.v. FiloSoft, Postbus 1353, 9701 BJ Groningen, of telefonisch (050-137 746).

VERENIGINGEN EN CLUBS

NEDERLANDSE VERENIGINGEN & CLUBS

HCC 68000 GG
 Secr: R.N.Smink
 Waterlelie 9
 2851 ZL Haastrecht (01821-2491)

HCC SINCLAIR GG
 Secr: T.A.P. Molenaar
 Postbus 76
 2260 AB Leidschendam (01670-66845)

Sinclair Gebruikersgroep Groningen
 Secr: A.Westenberg
 Bottemaheerd 48A
 9737 NG Groningen (050-424136)

Spectrum Gebruikers Groep Hoogeveen
 Secr: G.Guchelaar
 Neptunus 16
 7904 GE Hoogeveen

Spectrum Gebruikers Groep Eemmond
 Secr: G.J.Kuipers
 Graaf Edzardstraat 65
 9902 HK Appingedam (05960-23890)

Sinclair Gebruikersgroep Ter Aar
 Secr: Peter Bey
 Sperwerhoek 36
 2743 GC Waddinxveen (01828-19975)

Sinclair Gebruikersgroep Leeuwarden
 Secr: Fransjan M.Punt
 Hesteromhei 21
 9254 EC Hardegarijp (05110-5840)

Sinclair Gebruikers Groep Amsterdam
 Postbus 61289
 1005 HG Amsterdam (020-880126)

Sinclair Gebruikers Groep Assen
 Secr: C. van Krimpen
 Koldakker 34
 9407 BM Assen

Dwarsligger Computer Club
 Systeem Groep Sinclair
 Secr: J. van der Zee
 Pleiadenplantsoen 60
 1973 BW IJmuiden (02550-10002)

Sinclair Gebruikers Groep Den Bosch
 Secr: Ad van Liempt
 Bingerden 6
 5235 DM Den Bosch (073-416837)

Spectrum Gebruikers Groep Dordrecht
 Secr: R.Uitterlinden
 Noorderdijk 379
 3312 AG Dordrecht (078-312262)

Systeem Gebruikers Groep Apeldoorn
 Postbus 1061
 7301 BH Apeldoorn

Discovery User Club
 Secr: D.C.Kruithof
 Boeierskade 6
 2725 CH Zoetermeer (079-416360)

Sin-QL-air
 Secr: B.Visser
 Scheepmakerskade 30
 3011 VX Rotterdam (010-143554)

QL-groep Noord-Nederland
 Secr: R.Russchen
 de Biskop 6
 9243 WG Bakkeveen (05169-1016)

Nederlandse Genealogische Vereniging
 Afd: Computerdienst
 Contact voor Sinclair Gebruikers:
 K.B.Scholten
 Hollandshof 84
 2411 KH Bodegraven

BELGISCHE VERENIGINGEN & CLUBS

Sinclair Gebruikers Groep Aalter
 Secr: Weibroekdreef 71
 9800 Aalter (091-742205)

Spectrum Gebruikersgroep Antwerpen
 Secr: Boshovenstraat 123
 2100 Deurne (03-3247294)

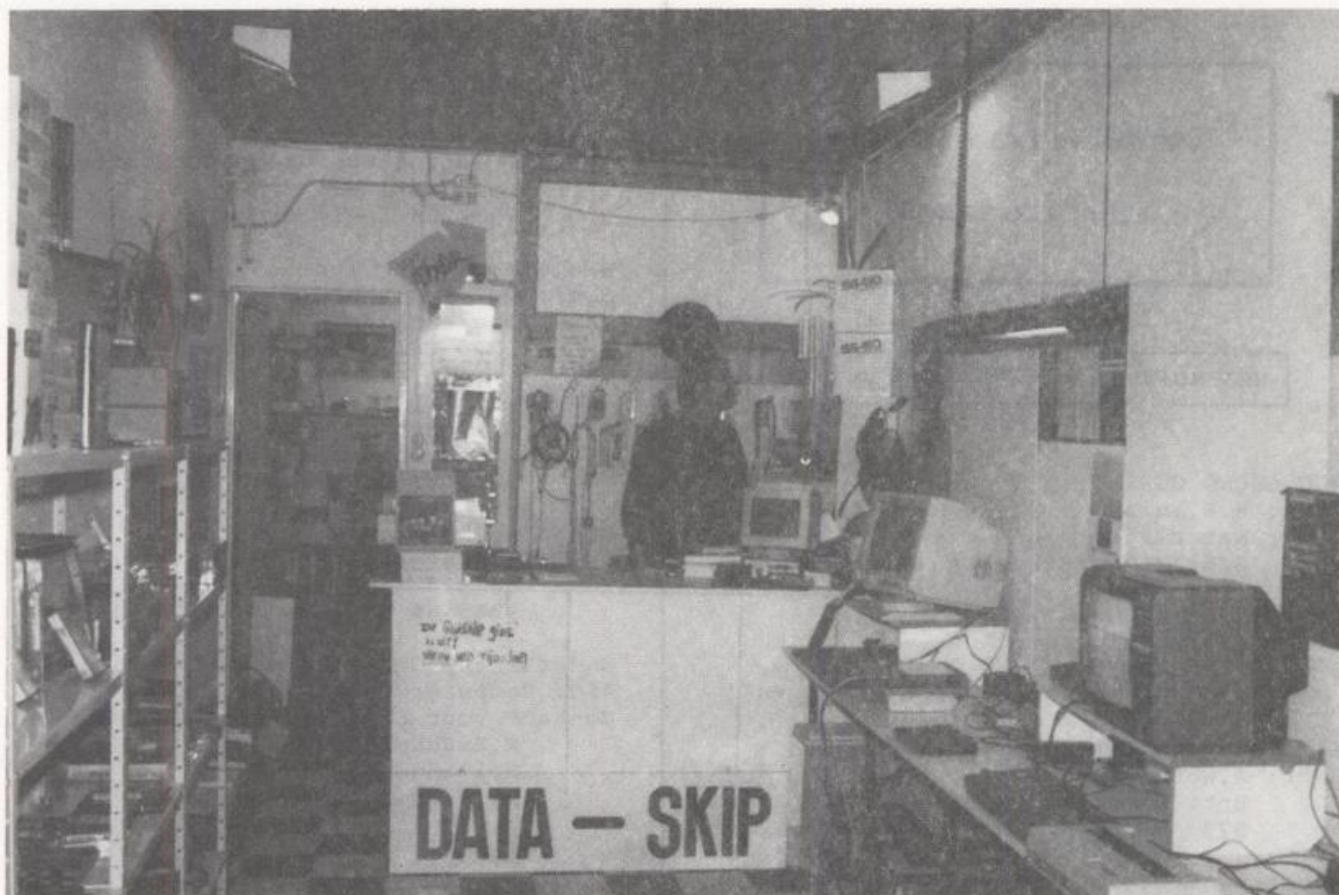
Sinclair Gebruikers Groep Gent
 Secr: Zwijnaartsesteenweg 148
 9000 Gent (091-218786)

Spectrum Gebruikers Groep Lier
 Secr: Baron Carolylaan 8
 2500 Lier (03-4803448)

Sinclair Groep Limburg
 Secr: Oude Kuringerbaan 65
 3500 Hasselt (011-251628)

QL-Gebruikersgroep Limburg
 Demerstraat 17
 3500 Hasselt (011-227213)

Multi User Club
 Secr: J. De Plancke
 Astridlaan 25
 8320 Brugge Assebroek



Data - Skip

Vanaf de markt in Gouda is de Lange Willemsteeg gemakkelijk te vinden. Op de markt vind je een stadsplan. Kleine vijf minuten lopen, en je bent er. Geen reusachtige etalage met glitter en lichtjes, maar een 19de-eeuws huisje, met kleine vriendelijke raampjes, volgestouwd met alles en nog wat, dat op Sinclair computers aangesloten kan worden. Je proeft duidelijk de hobby-sfeer. Hier wordt nog geknutseld, dat is duidelijk.

We worden ontvangen door Ton Versluis, de eigenaar van de tweemans-zaak. Breed glimlachend leidt hij ons rond in zijn kleine koninkrijk. In april 1985 is het allemaal begonnen. Niet - zoals enkelen van zijn collega's - op een zolderkamer, maar in de hall van zijn huis, naast de huidige zaak. Zijn enthousiasme over z'n eigen Spectrum zette hem aan om zich in Sinclair producten te specialiseren.

De oorspronkelijke doelstelling was, om goede nederlandse software tegen een redelijke prijs op de markt te brengen. Dat viel wat tegen : het aanbod was zeer gering, en de kwaliteit van de nederlandse software was matig, in vergelijking met wat er uit Engeland aan kwam spoelen. Bovendien ondervond DATASKIP al gauw dat software uitgeven erg bewerkelijk is. Je blijft aan de gang : advertenties opmaken, hoesjes ontwerpen en produceren, handleidingen schrijven ...

Daarom werd in september 1985 ook met hardware voor Sinclair begonnen. Niet meteen in grote hoeveelheden : hardware vergt grote investeringen. Toch is de zaak vrij snel uitgegroeid. Momenteel levert DATA-SKIP een breed gamma van Sinclair-artikelen : printers, beeldschermen, software allerlei, boeken en diverse interfaces. Het grote deel daarvan importeert DATA-SKIP zelf.

Van toen af begon de ontwikkeling erg snel te gaan. Sinds begin 1986 levert Data-Skip ook aan andere winkels. Door de groeiende omvang van de eigen import uit Engeland werd er meer en meer aan distributie gedaan.

Dat was nog niet genoeg. DATA-SKIP begon ook nog zelf met de ontwikkeling van randapparatuur. In augustus 1986 kwam de Videoface digitiser op de markt. Een uitgebreide bespreking van dat apparaat vind je in nummer 2 van dit tijdschrift.

Deze digitiser kent een enorm succes : vanuit Australië, Hong Kong, Uruguay en Saoedi-Arabië komen reacties en vragen om informatie. Zelfs in het anders zo chauvinistische Good Old England loopt het ding als een trein. Dat is wel een grote pluim op de hoed van de producent. Het is niet bepaald eenvoudig om als Nederlands bedrijf op de Engelse markt door te dringen.

Sinds kort levert DATA-SKIP ook de muis van AMX (zie nummer 2). Omdat we dit apparaatje zo deksels handig vinden, hebben we besloten om in samenwerking met DATA-SKIP een programmeer-wedstrijd voor die muis te organiseren. Meer daarover vind je elders in dit nummer. Er zijn mooie prijzen te verdienen, dit kunnen we al vertellen! DATA-SKIP werkt momenteel aan eigen nederlandse producties rond de AMX-muis. Dit kunnen we alleen maar toejuichen: er is veel te weinig goede nederlandse software op de markt.

Daarnaast staan er nog op stapel: een sideway-ROM (prik die op de achterkant van je Spectrum en je hebt ogenblikkelijk toegang tot de programma's die in die ROM staan) en een nieuw inbouwsysteem in de aard van de THOR, maar dan voor de Spectrum, ter vervanging van de inmiddels ter ziele gegane Opus. Dat laatste is nog in het projekt-stadium, maar DATA-SKIP kennende zullen we er nog meer over horen.

DATA-SKIP heeft een duidelijke filosofie over zakendoen. Hun grote sleutelbegrip (keyword?) is: klantvriendelijkheid. Alle binnenkomende programma's worden nauwgezet bekeken en uitgetoetst. Dat maakt, dat de klanten een gefundeerd advies krijgen over wat voor hem/haar de beste oplossing voor de gestelde vraag



kan zijn. Dit geldt ook voor hardware. Elk stuk wordt getest, zodat alles wat de deur uitgaat, ook gegarandeerd werkt. Dat maakt de kans op stukken natuurlijk minimaal. Dat is prettig voor de klant, en dat brengt ook mee dat de eigen reparatiedienst meer aandacht kan besteden aan dingen die dan toch fout gelopen zouden zijn.

DATA-SKIP gaat er over het algemeen van uit dat Sinclair machines niet bepaald de duurste computers zijn, dus dat je ook met randapparatuur in een redelijke prijsklasse moet blijven. Daar zijn wij het natuurlijk roerend mee eens.

Die hele politiek heeft er ondertussen voor gezorgd, dat DATA-SKIP in de korte tijd van zijn bestaan, toch al een ruime klantenkring heeft opgebouwd, zowel in Nederland als in België.

Het klinkt als een cliché, maar toch: wij zijn ervan overtuigd dat de vriendelijke prijzen plus het vriendelijke gezicht van DATA-SKIP hebben bijgedragen tot het snelle succes van dit bedrijf. Ton Versluys blikt overigens met veel vertrouwen naar de Sinclair-toekomst. Volgens hem is de Sinclair-markt nog zeker goed voor vijf jaar gezond werken. Alles hangt natuurlijk af van de samenwerking van vier factoren: Sinclair-dealers, hobbyisten, ondersteuning via een tijdschrift, en ene meneer Sugar. Voor de derde poot zorgen wij alvast.

Wie zich met eigen ogen wil overtuigen, kan terecht op maandag vanaf 13.30 tot 16.00 uur, en tot en met zaterdag tussen 10.00 en 16.00 uur, of later dit jaar op de ZX Microfair in Londen, op de eigen stand van DATA-SKIP.

Data-Skip
L. Willemsteeg 10, 2801 WC Gouda.
Tel. 01820-20581



Speciaal voor de lezers van de SINCLAIR GIDS is er deze cassette service. In eerste instantie betreft het hier verzamelcassettes met programma's uit de SINCLAIR GIDS.

Naast de verzamelcassettes zullen er ook nog zgn. SPECIALS gepubliceerd worden. Dit zijn cassettes die een speciaal programma bevatten en behoren tot de zogeheten "LOW-COST"-software. Dus goede programma's tegen een redelijke prijs.

Als er voldoende belangstelling is zullen ook voor de ZX81 en de QL resp. verzamelcassettes, -cartridges verschijnen. Dit laatste is uiteraard afhankelijk van het feit of er wel voldoende programma's voor die computers bij de redactie binnenkomen (zie de rubriek LLIST).

De volgende cassettes zijn inmiddels leverbaar:

SG-01-S Verzamelcassette met de programma's uit SINCLAIR GIDS no.1+2.
Spectrum fl.17,50/350 BF
Basisabonnee : fl.12,50/250 BF

SPECTRUM SPECIALS:

SS-01 Special 1: TT-S Toolkit voor programmeurs incl. Ned.Handleiding. De GAMMA-TOOLKIT met 10 extra commando's: TRACE, FIND, RENUMBER, MOVE, etc. verder een UDG-ontwerper, een RAM-test-programma, SCREEN tekenprogramma en TAPE-INVESTIGATOR. 5 Utilities voor...fl.27,50/550 BF
BASIS-Abonnee: fl.22,50/450 BF

SS-02 Special 2: ASTROLOGIEPROGRAMMA voor slechts:.....fl.15,00/300 BF
BASIS-Abonnee: fl.10,00/200 BF

SS-03 Special 3: Utilities 1: COPY-A4, COPY RS232, MCOPY. Screen copy-progs voor Centronics en RS232 en een M-Drive copy-prog. 5 utilities voor...fl.15,00/300 BF
BASIS-Abonnee: fl.10,00/200 BF

SS-04 Uitverkocht

ZX 81 SPECIALS:

ZS-01 QSAVE laadt/savet tot 16x sneller voor ZX81 16K-64K..fl.25,00/500 BF
BASIS-Abonnee: fl.20,00/400 BF

ZS-02 MCODER vertaalt Uw BASIC in M/C voor ZX81.....fl.15,00/300 BF
BASIS-Abonnee: fl.10,00/200 BF

QL-SPECIALS:

QL-01 MACRO-ASSEMBLER van GST systems met screen editor van Metacomco. was fl.275,00 nu:fl.165,00/3300BF
BASIS-Abonnee: fl.140,00/2800BF

QL-02 QL-Toolkit 1 van Tony Tebby
nu voor:.....fl. 90,00/1800BF
BASIS-Abonnee: fl. 79,00/1580BF

BOEKEN voor QL:

The sinclair QL series van Hutchinson.
Originele prijs fl. 39,00 per stuk
QL-B1: Introduction to Superbasic on the Sinclair QL nu:fl. 12,50/250 BF
QL-B2: Machine Code programming on the Sinclair QL nu:fl. 12,50/250 BF
QL-B3: Advanced programming on the Sinclair QL nu:fl. 12,50/250 BF
QL-B4: Database management on the Sinclair QL nu:fl. 12,50/250 BF
QL-B5: Word processing with the Sinclair QL nu:fl. 12,50/250 BF
QL-B6: Desk-top computing with the Sinclair QL nu:fl. 12,50/250 BF
BASIS-Abonnees deze QL-Boeken :fl. 10,00 /200 BF.

BOEKEN ZX SPECTRUM:

ZS-B01 PROGRAMMING COURSE for the ZX-Spectrum (500+ pagina's !!)
nu: :fl. 20,00/400 BF
BASIS-Abonnee :fl. 17,50/350 BF

Alle prijzen incl.BTW en verzendkosten !
AANBIEDINGEN gelden zolang de voorraad strekt.

Bestellen door overmaking (of betaalcheque/girobetaalkaart) op giro 5109074 t.n.v. de SINCLAIR GIDS, Baarle Nassau. België: Postrek: 000-1592677-34 tnv. Sinclair Gids, Baarle Nassau. Betalen bij ontvangst kan ook, maar dan worden de rembourskosten(fl.8,75/165 BF) extra in rekening gebracht.

VERMELD BIJ ALLE BESTELLINGEN DE NUMMERS VAN DE CASSETTES WAAR HET OM GAAT !!

(VERVOLG VAN BLZ. 4)

8- Maak je ook geen zorgen over de opmerkingen die je naar je hoofd krijgt. Men is misschien jaloers.

Op een gegeven moment merk je, dat je heel wat zekerder wordt in de buurt van een computer. Je begrijpt ook de computervaktaal een beetje beter.

En laat je niet van de wijs brengen door personen die zeggen dat je niets van computers weet, omdat je maar zo'n speelgoed-dingetje hebt. Zij hebben namelijk al veel eerder met zo'n ding te maken gehad, toen ze op de computer wegwijzen werden gemaakt. Ze zullen het nooit toegeven, maar vijf jaar geleden was hun "computer-op-de-zaak" niets meer (eerder minder) dan de tegenwoordige "speelgoed-computer". En ze waren toen onbetaalbaar duur.

En wees nooit bang om fouten te maken. Daar leer je van ! Ineke Maas (vrouw)

The great Escape

Het is 1942, je bevindt je in een Duits krijgsgevangenkamp. Uiteraard wil je ontsnappen. Het kamp is gevestigd in een oud kasteel op een afgelegen plaats in de bergen en slechts bereikbaar via een smal bergweggetje.

Het kamp wordt zwaar bewaakt met behulp van honden en overal staan uitkijktorens. 's-Nachts wordt er gepatrouilleerd en sterke zoeklichten speuren langs de muren. De gevangenen zijn ondergebracht in barakken op de binnenplaats van het kasteel. Bij het kasteel is een gebied waar de gevangenen kunnen recreëren. Daar bevindt zich ook de goed verborgen toegang tot het onderaardse gangenstelsel dat de gevangenen in het geheim hebben aangelegd.

Het spel: Je moet proberen via dit gangenstelsel uit dit kasteel te ontsnappen. Je moet daarbij wel even een aantal zaken goed in de gaten houden. Allereerst moet je er achter zien te komen, hoe de dagindeling is. Als je dat weet, kun je op gezette tijden een poging doen om te ontsnappen. In de onderaardse gangen is het echter zonder toorts gevaarlijk, je zou gemakkelijk kunnen verdwalen. Het is wel een ideale plek om allerlei hulpmiddelen te verbergen.

Links op het scherm staat een vlaggestok met een vlag, die je moraal aangeeft. Deze gaat omlaag als je gepakt wordt op illegaal terrein (je bent op illegaal terrein als de vlag rood kleurt) of als er verborgen spullen van je ontdekt worden. Op het laatst is je moraal zo gebroken, dat je geen lust meer hebt om nog een poging tot ontsnappen te ondernemen. Het mannetje op het scherm kun je dan ook niet meer in beweging krijgen. Als je een tijdje niets doet terwijl de vlag groen is, gaat het mannetje op het scherm weer in het gareel lopen en doet precies hetzelfde als de andere gevangenen. Op deze manier kun je eenvoudig door NIETS te doen achter de dagindeling binnen het kamp komen. Dan kun je uitvinden op welke tijden je ongemerkt kunt wegslopen om op onderzoek uit te gaan.

Grafisch ziet het spel er zeer verzorgd uit. Je waant in een echt kasteel. Dit spel is z'n geld dubbel en dwars waard.

(E.S.)

Titel: The Great Escape
Medium: Cassette (48K Spectrum)
Productie: Ocean Software
Prijs: fl. 36,00 / 720 BF
Import: Nederland: Dataskip, Gouda
Belgie: Micro-Connection, Antwerpen

NIEUW NIEUW NIEUW

HANDBOEK VOOR SPECTRUM 128K PLUS 2

De Nederlandse editie van het uitgebreide handboek voor de ZX SPECTRUM PLUS 2.

130 pagina's dikke handleiding voor het in BASIC programmeren van de 128K versie van de beroemde ZX SPECTRUM (+). Bevat alle informatie die U nodig heeft om zelf programma's te schrijven.

Vele voorbeelden ondersteunen de uitleg van de keywords en functies.

Eindelijk weer een standaardwerk over een Sinclair Computer met alle benodigde informatie zowel over het werken in de 48K-mode als werken in de 128K-mode.

Uitgebreide technische informatie over de geheugenstructuur, programmaopslag in het computergeheugen en alle SYSTEEM-VARIABLEN.

Een onmisbaar handboek voor een ieder, die met een 128K Spectrum werkt en tot op heden de goede documentatie over deze voortreffelijke machine moest ontberen.

Dit handboek zal medio februari 1987 verschijnen en zal verkrijgbaar zijn in de goede boekhandel, de Sinclair Speciaal-zaken en het Computercollectief.

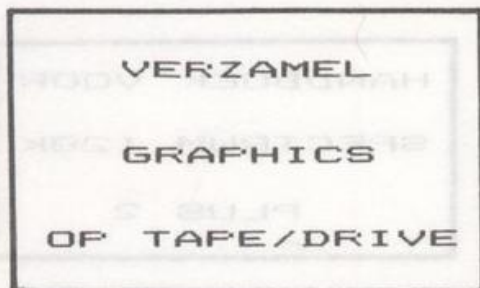
ISBN 90-6883-028-7

fl. 32,90 / 658 F

Een uitgave van:

TERMINAL SOFTWARE PUBLICATIES
Postbus 111,
5110 AC Baarle Nassau.

NIEUW NIEUW NIEUW



Het handige bij Tasword 2 is, dat je de 16 grafische karakters van de Spectrum kunt gebruiken om diverse controlecodes voor je eigen printer op voorhand te definiëren. Die tekens kun je dan in je tekst intypen en Tasword zorgt ervoor dat de printer de correcte codes krijgt toegestuurd.

Je kunt bijvoorbeeld de "TRUE 8" (het lege vierkantje) definiëren als code 12. Dat is voor de meeste printers het sein om het blad waarop ze aan het printen waren, helemaal eruit te draaien. Zet je dan op een bepaald punt in je tekst, waar je een nieuwe bladzijde wilt beginnen, zo'n teken, dan krijg je precies wat je wilt. Je hoeft dan niet zelf het printen te onderbreken en het papier verder te draaien. Je moet ook niet bij het begin opgeven dat regels zo tot zoveel geprint moeten worden. Daar zorgt dat ene grafische teken voor.

Je kunt ook de "INVERSE 8" (het zwarte blokje) definiëren als code 10. De meeste printers begrijpen die als "draai een regel verder zonder te printen". Door dit teken te gebruiken, kan je een reeks lege regels uit je tekstfile halen, waardoor je meer tekst in een file kwijtkan.

Het vervelende aan Tasword 2 is, dat de grafische tekens die je gedefinieerd had verloren gaan wanneer je de computer uitschakelt. Wanneer je dezelfde tekst daarna nog eens wilt afdrukken, moet je dus weer opnieuw die tekens definiëren. Dat gaat op de duur vervelen.

Daarom dachten wij, dat het wel handig zou zijn, indien je een bepaalde set graphics voor een bepaalde tekst, na het definiëren ook op tape/disk kon bewaren. Dan hoef je, indien je een bepaalde tekst wilt gaan afdrukken, enkel nog vooraf de set graphics die bij die tekst horen, in het geheugen te laden, en je kunt gaan printen. Makkelijker kan nauwelijks.

De aanpassing die daarvoor in de BASIC van Tasword 2 moet gebeuren, is niet eens zo omvangrijk. Drie extra regeltjes BASIC en de zaak is klaar.

Regel 300 bereik je, indien je de letter "G" indrukt, wanneer je het BASIC-menu ziet. Gewoonlijk loopt het programma dan door op regel 305. Tussen die twee regels moet dus alles gebeuren.

We voegen een regel 301 tussen, waarop we de keuze krijgen tussen : een set inladen, een set wegschrijven, of een nieuwe set aanmaken. Dat kan op een simpele manier :

```
301 INPUT "LOAD / SAVE / NIEUW
";A$: IF A$="N" THEN GO TO VAL "
305"
```

Nu hebben we onze keuze gemaakt. Indien we "N" antwoordden, dan loopt alles zoals voor de aanpassing. Gaven we "L" of "S" in, dan moet er iets anders gebeuren. Daarom schrijven we een regel 303, waarop we een CAT opvragen van de drive, en vervolgens een bepaalde set graphics in het geheugen laden.

```
303 CAT md: INPUT "NAAM ? ";N$:
IF A$="L" THEN LOAD "*"M";MD;N$CO
DE : GO TO VAL "22"
```

Uit de "IF" op die regel, moet duidelijk zijn dat, indien we niet "L" hadden ingegeven, de andere optie op de volgende regel wordt uitgevoerd :

```
304 SAVE "*"M";MD;N$CODE VAL "60
860",VAL "64": VERIFY "*"M";MD;N$
CODE : GO TO VAL "22"
```

De minder gelukkigen onder de lezers, die tekstverwerking met cassette doen - en dat kan natuurlijk ook, waarom niet - moeten natuurlijk wel een en ander wijzigen aan de vorige regels. Het CAT-commando hoort er niet, en de syntax van de LOAD- en SAVE-commando's is anders. Voor het overige is alles gelijk. Het enige ongemak is, dat de cassette-werkers de positie van de diverse sets op de cassetteband ofwel uit het hoofd moeten kennen, ofwel op een papiertje bij moeten houden.

Overigens, nu we het over cassettes hebben : ikzelf maak er een gewoonte van om van de definitieve versie van alle belangrijke tekstfiles, een back up te maken op een cassette. Mijn ervaring heeft me geleerd dat, hoe ingenieus en handig ze ook zijn, microdrives altijd de belangrijkste files na een tijdlang niet meer kunnen terugvinden op een cartridge. De wet van Murphy heeft blijkbaar ook vat op spijstechnologie...

Nu vond ik het wat onpraktisch om elke keer ik een back up wou maken op cassette, de BASIC van Tasword 2 aan te passen. Daarom heb ik ook hiervoor een aanpassing in de tekst gemaakt, waardoor ik altijd de keuze heb, zowel bij SAVEN als bij LOADen, om met het ene of het andere medium te werken.

Terwijl ik toch aan het veranderen was, bedacht ik dat het ook vaak gebeurt dat ik achteraf aan een tekst nog een paar wijzigingen of correcties moest doen, en dat de microdrive het vertikt om een file te SAVEN onder een naam die al op die cartridge staat.

Ook de Opus diskdrive "sjoemelt" daar mee: die accepteert wel dezelfde naam, maar wist de file niet echt van de diskette. De nieuwe file wordt op de eerstvolgende vrije plaats geschreven, de oude naam wordt uit de "CAT"-file gewist, en de nieuwe naam wordt daaraan toegevoegd, met de andere gegevens (plaats op de schijf, lengte).

Opus splitst een file niet op, zoals de microdrive. Deze laatste vult gewoon elke lege sector die hij onder zijn leeskop ziet, op met een stukje van de file, tot ze helemaal op de cartridge staat. Opus wil de hele file netjes in een blok na elkaar hebben. Dat maakt dat een "CAT" je op een gegeven ogenblik vertelt dat je nog 50 K vrij hebt op de diskette, maar dat je, wanneer je een tekst op die diskette wilt zetten, de laconieke mededeling van Opus krijgt: "No room on disk"! Dat gebeurt niet, wanneer je de file eerst wist, tenzij de nieuwe tekst meer sectoren op de disk beslaat dan de vorige met dezelfde naam.

Om een en ander vlot te doen verlopen, dienen de nu volgende aanpassingen.

Voor het SAVEN van teksten:

```
1000 CLS : POKE 23658,8: INPUT "
DRIVE (D) OF CASSETTE (C) ? ";S$
: LET CAS=(S$="C")
1001 LET B=FN P(VAL "62216")
1002 IF NOT CAS THEN CAT MD: GO
SUB 1500
1003 INPUT "NAAM: ";A$
1004 IF NOT CAS THEN SAVE "M";M
D;A$CODE B,A
1005 IF CAS THEN SAVE A$CODE B,A
```

waarna het programma verder loopt op regel 1040 (er zijn geen regels tussen 1005 en 1040):

```
1040 CLS : PRINT "TEKST ";A$;" G
ESAVED: "'A;" BYTES, ";A/VAL "64
";" REGELS": IF CAS THEN VERIFY
"CODE : GO TO VAL "1110"
```

en indien er niet met cassette gewerkt wordt, volgt regel 1100 met:

```
1100 VERIFY "M";MD;A$CODE B,A
```

De korte subroutine vanaf regel 1500 zorgt voor het wissen van de gewijzigde tekst:

```
1500 INPUT "WIS FILE ?";A$: IF A
$="" THEN RETURN
1505 ERASE "M";MD;A$:RETURN
```

Een gelijkaardige aanpassing laat de keuze bij het inladen van teksten. Dit gebeurt vanaf regel 2000.

```
2000 CLS : POKE 23658,8: INPUT "
DRIVE (D) OF CASSETTE (C) ? ";S$
: LET CAS=(S$="C"): IF NOT CAS T
HEN CAT MD
2010 INPUT "NAAM: ";A$
2030 LET B=FN P(VAL "62216"): IF
NOT CAS THEN LOAD "M";MD;A$CODE
(A+B),((FN P(VAL "62221")+VAL "
22")*VAL "64"-A): GO TO VAL "10"
2040 LOAD A$CODE (A+B),((FN P(VA
L "62221")+VAL "22")*VAL "64"-A)
: GO TO VAL "10"
```

Regels 1000 en 2000 zorgen er voor, dat je door gewoon "ENTER" te drukken, als antwoord op de vraag "DRIVE/ CASSETTE", automatisch voor drive kiest. Dat kan ook andersom, door op beide genoemde regels de logische uitdrukking (S\$="C") te vervangen door (S\$<>"C").

Om zo weinig mogelijk extra typwerk te krijgen, door de aanpassingen, zorgt regel 1500 er ook voor dat gewoon "ENTER" drukken als antwoord op de vraag of er een tekst gewist dient te worden, voldoende is om niets te wissen.

We hopen dat uit de reeks van drie artikelen over Tasword 2, duidelijk is geworden hoe flexibel dit programma wel is. We gooien de handschoen nu in jullie kamp, voor nog meer nuttige aanpassingen van Tasword 2, of zelfs Tasword 3: we kunnen ons niet voorstellen dat welk programma ook, echt in alle eisen en behoeften van zijn gebruikers kan voorzien. Schrijven maar! (P.P.)

(vervolg van blz. 14)

Echt h le hoge geluiden pikt de sampler niet. Nu is 10 kHz niet laag, maar de boventonen van geluiden bepalen wel de "kleur" ervan. De menselijke stem wordt bv. heel natuurgetrouw weergegeven; de hoge noten op een piano nog nauwelijks.

Wat we niet getest hebben, bij gebrek aan het nodige materieel, is het gebruik van de Music Machine in samenwerking met een MIDI-keyboard. Volgens de handleiding moet daar zowat alles mee mogelijk zijn. Het keyboard kan alle samples gebruiken, op zichzelf of gemixt met wat de synthesizer zelf bevat. Alle 16 MIDI-kanalen kunnen met drums of muziek van de Music Machine worden verbonden.

Ik denk dat de Music Machine een handig hulpmiddel kan zijn voor elke Spectrum-gebruiker die zich met elektronische muziekmakerij bezig houdt. Het is geen Stevie Wonder-machine. Maar voor 3990 fr (ca. f.199.-) kan je dat ook niet eisen.

Belgi  : Micro-Connection,
St.-Katelijnevest 16-18
2000 Antwerpen
tel. 03/231 01 03

Nederland :

Data-Skip
L. Willemsteeg 10
2801 WC Gouda
tel. 01820-20581

Komin BV
de Greefstraat 15A
5622 GJ Eindhoven
tel. 040-456660

HELP !! Mijn Z80A-PIO-chip werkt niet op mijn 128K Spectrum. Wie geeft mij de oplossing ? R.Boegheim. Tel:01820-26272.

IK ZOEK: Een 48K SPECTRUM zonder software en hardware. Een ECHTE Speccie dus!!
Tel:023-255662 na 18.00 uur

IK ZOEK:Een kraakinterface voor de ZX Spectrum. Reacties aan: J-R Broeksma, Buitenes 4, 9407CN Assen Tel:05920-45005

IK ZOEK: Een goed werkend boekhoudprogramma voor prive-gebruik. Wie kan mij helpen ? Ben Haitmaa, Harste 7,8602 JX Sneek.

IK ZOEK: TELEX/MORSE programma voor spectrum met en zonder converter.
H.Snijder, Dwingeloo Tel:05219-2285

IK ZOEK: contact met gebruikers van programma voor "STAMBOOMONDERZOEK".
E.H.Eppens, van Linschotenstraat 31, 9601 HH Hoogezand.Tel:05980-93179

WIE kan mij het een en ander vertellen over de GLP PRINTER. WIE heeft er ervaring mee ?
K.H.Schaap,Zwaluwstr 8,2224 HN Katwijk

IK ZOEK:ANIMATOR 1 evt. ruilen voor THE COLT (compiler). Reacties aan:
E.Groot Zwaartink, K.Nilantstraat 157, 7415 TC Deventer.

IK ZOEK: Euroelectronics LPRINT III Interface.
Tel: 05170-7690

WIE kan mij helpen bij het aanpassen van het programma MASTERFILE voor printer.
R.Verhoeff,Teunisbloemplein 31,Den Haag

IK ZOEK:6EBRUIKERS van de AMX MUIS om er het uiterste uit te halen. Schrijf naar:
R. v.d. Bunt,Klokkegat 10, Lunteren.

IK ZOEK:TEGEN VERGOEDING VAN KOSTEN: het boek: "The Working Spectrum" door: D. Lawrence.
Mike Rolff, Aardenburg. Tel:01177-2416 na 17.00 uur.

GEZOCHT: Gebruikers van de 128K Spectrum die weten hoe je "Stargliders 128K" op een OPUS-DISK kan zetten.
Tel:078-60096

WIE kan mij helpen aan de handleiding van ASTRONOMER ? Kosten worden vergoed.
A.G.Mertens. Tel:078-174357

IK ZOEK: 1 Microdrive + evt. cartridges (Niet te duur).
M.Karel, Reinaartpad 12, 3813 KM Amersfoort (033-756196).

IK ZOEK: De tweede diskdrive voor de OPUS Discovery. Bij voorkeur de 718K versie, maar ook de standaard.
Tel: 010-4348089

IK ZOEK: Duidelijke beschrijving voor uitbreiding 48K naar 80K (Spectrum).
H.J.Nijdam, Fr.Hendriklaan 46, Haarlem. (023-280436)

IK ZOEK: Van YOUR SPECTRUM de nummers 1 t/m 13. E.Vermeulen (010-4709918) na 18u

IK ZOEK: MARGAMES voor ZX Spectrum.
W.Jongejan, Tel: 03480-18586 voor 1800u.

TE HUUR GEVRAAGD: Multiface 1 voor backup van mijn software naar microdrive.
H.Meussen. Tel: 05208-66832

IK ZOEK: Module Digitale Geheugen-oscilloscoop voor Spectrum met software.
Tel: Belgie: 013-314846 na 1800 uur.

IK ZOEK:kopieen van ZX Computing over de "Venturespeakeditor". Rob Udo, Neptunusweg 32, 5632 CP Eindhoven

IK ZOEK: Originele programma's voor de Spectrum 128K. C.K.Man, Westerstraat 16, 1441 AS Purmerend.

IK ZOEK: Iemand om originele software te ruilen. Ik zoek 'n goed Nederlands aardrijkskundig programma. Bel: 05709-1269

IK ZOEK: een ZX80 met eventuele uitbreidingen. J.Jongkind, A.H.Blaauwstr. 13, 1431 KT Aalsmeer. (02977-25812)

TE KOOP:Originele 128K Spectrum Software o.a. 'Robin o/t Wood', 'Ghostbusters' enz
Tel: 078-60096

IK BIED AAN: ZX LPRINT 3 Interface voor f1. 150,00 of f1. 195,00 met kabel.
Tel: 053-353723 vraag naar Jan-Remco.

TE KOOP:PSION ORGANISER XP f1.400/7200BF Cartridgebox met 26 cartridges voor 2800BF.
J. Peersman, Tel:Belgie:03-2368096

ABONNEMENT IS VOORDELIJG
 met een abonnement op de SINCLAIR GIDS
 krijg je 5 nummers voor fl. 45,00

Port Betaald
 Port Payé
 Baarle-Nassau

sinclair gids

Postbus 111
 5111 AC Baarle-Nassau

GRATIS LEZERSERVICE
 voor onze lezers

Zoekt u iets of heeft u iets aan te
 bieden?
 Plaats dan een annonce in de rubriek
 IN/OUT.

IK BIED AAN:

Port Betaald
 Port Payé
 Baarle-Nassau

sinclair gids

Postbus 111
 5111 AC Baarle-Nassau

in/out

TE KOOP: Spectrum 16K + 2 cassettes + 4
 boeken + voeding: fl. 200,00.
 Jan Helling, Mooiberg 10, 5114 AH
 Castelle.

TE KOOP: ZX Spectrum + Interface 1 +
 microdrive + joystickinterface + joycard
 + lichtpen + software voor: fl. 550,00.
 Tel: 01718-23579 (Hans).

TE KOOP: 8-bits I/O-kaart voor ZX
 Spectrum. Ook geschikt als joystickin-
 terface. Tel: 04750-21018

IK BIED AAN: Seikosha GP-50-S en een ZX
 Printer en Currah Speech Synthesizer.
 T.e.a.b. W. Duivenboden, Den Haag
 Tel: 070-675921

TE KOOP: ZX Spectrum 48K met kabels,
 voeding, software en 5 boeken.
 Vraagprijs fl. 125,00. Tel: 04499-2576
 na 1800 uur. (Susteren)

TE KOOP: Seikosha GP50S Printer, alleen
 papiertransport-motor defect.
 Tel: 01184-18677

TE KOOP: Voor Spectrum 48K(+):
 Microdrive+Interface I Fl. 150,00
 10 Microdr. cartridges+ opbergdoos fl.40
 Kempston Joystick Interface fl. 25,00
 Programmeerbare Joystick Interface fl.40
 Multiface One (backup naar microdrive):
 fl. 75,00. Boeken: Werken met ZX Micro-
 drive fl. 12,50. Spectrum Microdrive
 Book (Engels): fl. 12,50.
 J.J. Leune, Vignolastraat 124, Rotterdam
 Tel: na 19.00 uur: 010-4555665

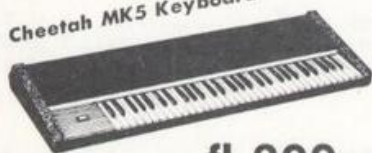
DATA-SKIP UW SINCLAIR-GIDS

Software, games

Xevious*	fl 39,-
Gauntlet*	39,-
1942	32,-
Lighthforce	32,-
Academy	39,-
Space Harrier*	32,-
Terra Cresta*	32,-
Sam. Fox strippoker	36,-
Bom 6 Jack	29,-
Strike Force Harrier	39,-
PSI Chess, 3 D*	45,-
Thamatos*	39,-
Graphic Adr. Creator	95,-
Konami's Coin-op Hits*	45,-
Elite Hit-pak	45,-
They-sold-a-milion	45,-
Explorer*	32,-
Starglider*	59,-
* is nieuw	

Aanbieding 1

Cheetah MK5 Keyboard



fl 399,-

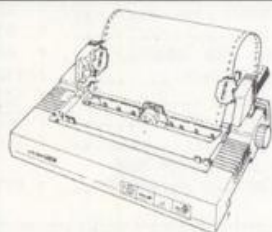
Midi interface

48/128/+Z fl 125,-

Software, serieus

Tasword III, cartr.	fl 69,-
Tascopy, screendump	45,-
Tasprint, 6 fonts	45,-
Tas-diary, dagboek	45,-
Mini-Office	29,-
Omnicalc	69,-
Skip-64, 64 kolom	19,-
Art Studio	65,-
Masterfile	69,-
Beta Basic	69,-
Laser Genius	65,-
Laser Basic	65,-
Laser Compiler	65,-

Aanbieding 2



Smith Corona par.-printer

fl 399,-

Aanbieding 3



ZX-Spectrum Plus Twee

fl 649,-

Software, 128 K.

Art Studio 128 k	89,-
Star Glider	59,-
Winter Games	39,-
Knight Time	19,-
3-weeks in paradise	29,-
Music Box	45,-
Tasword 128	65,-
Zub	19,-
Glider Rider	39,-
Samanta Fox Strip	39,-
Rasputin	36,-
Fairlight	36,-
Thanatos	39,-
Gladiator	36,-
Laser Genius (Assembl.)	65,-
Anal. of Rome	39,-

Stofhoezen

Spectrum plus	19,-
Spectrum 128 k	19,-
Sinclair QL	19,-
Saga Elite	24,-
Saga 2+	24,-
Saga Emperor	24,-
Lo Profile	24,-

Aanbieding 4



Compleet modern
pakket voor Sinclair
Q.L. Incl. auto-dial en
auto-answer.
Elke QL-bezitter start
nu z'n eigen databank
voor maar

Tandata
fl 325,-

Printers

Centronics G.L.P. een 80-koloms
printer voor normaal papier. Met
ser. en par. interface. Div. letter-
typen o.a. N.L.Q. 499,-
Citizen 120 D, schitternde par.
printer met vele mogelijkheden, o.a.
proportioneel, inverse, N.L.Q. 795,-

Monitors

Groen vanaf	299,-
Philips 7542, wit	369,-
Philips, kleur	695,-

Diversen

Seiko RC-1000, polshorloge
met o.a. opslag voor bijv. 30 stuks
adressen, telefoonnr's, formules e.d.
wereldtijden, alarm, 1 jaar vooruit
te programmeren
Voor gebruik let Interface I 109,-
Seiko QL software 29,-

Sinclair Q.L. hardware

Sinclair Q.L. compleet	495,-
CST Disk-interface	345,-
Modem-pakket, compleet	349,-
NLQ-printers vanaf	695,-
Monitors vanaf	299,-

Aanbieding 5



AMX-muis, incl. software

fl 249,-

ZX-Spectrum Hardware

Multiface One	175,-
Multiface One 128	199,-
Videoface Digitiser	249,-
Cartridge tox	19,-
VTX-5000 modem	199,-
ZXL-printinterface	199,-
3 1/2" diskette	5,-
Konix Speedking, joystick	39,-
Joystick Interface	49,-
Saga Elite, toetsenbord	275,-
Saga TNO-plus	225,-
Opus Discovery	595,-

POSTORDER: BEL 01820-20581
porto software fl 2,50
hardware fl 5,-
rembours fl 10,-

Prijswijzigingen voorbehouden
CST Thor bel voor inlichtingen
en prijzen

TEL 01820-20581
Data-Skip,
L. Willemsteeg 10
2801 WC Gouda

computercollectief

Amstel 312 (t.o. Carré) / 1017 AP Amsterdam / Giro 4 475 158 / Bank NMB 69.79.15.646

 * onze nieuwe WINTER 1986/87 CATALOGUS is nu uit. *
 * stuur ons een kaartje met je naam en adres en de *
 * vermelding 'SINCLAIR GIDS' en we sturen hem gratis toe. *

in BELGIE zijn al onze artikelen verkrijgbaar bij :
 Het Computerwinkeltje pvba,
 M Sabbestraat 39, B-2800 MECHELEN
 telefoon (015) 206 645

HIERONDER EEN OVERZICHT VAN NIEUWE EN ACTUELE BOEKEN VOOR DE SPECTRUM EN QL

BOEKEN voor de SPECTRUM

*Electronica Projecten voor de ZX Spectrum f 29,50
 Leren Programmeren ZX Spectrum+ boek 1 f 28,25
 Leren Programmeren ZX Spectrum+ boek 2 f 28,25
 *Leren Programmeren ZX Spectrum+ boek 3 f 28,25
 *Leren Programmeren ZX Spectrum+ boek 4 f 28,25
 een hele duidelijke serie programmeerboeken
 van Kluwer met beeldschermfoto's.

*BASICODE-3 boek & cassette f 27,50
 de cassette bevat 10 vertaalprogramma's,
 geschikt voor meer dan 30 microcomputers
 waaronder de Spectrum en de Spectrum+ en drie
 demonstratieprogramma's in BASICODE-3.

*Nederlandse Handleiding HISOFT Pascal -Spectrum f 15,00
 dit boek wordt alleen verkocht tegen inlevering
 van de 1e bladzijde van het Engelse Manual.

40 Grafische Programma's voor de ZX Spectrum .. f 29,50
 Ontdek de ZX Spectrum f 29,50
 Professionele Software voor de ZX Spectrum f 49,90
 Spectaculaire Spelen voor uw ZX Spectrum f 29,00
 100 Programma's voor de ZX Spectrum f 55,00
 BASIC Computerspellen voor de ZX Spectrum f 28,00
 BASIC-programma's voor de ZX Spectrum f 32,50
 BASIC-programma's voor ZX Spectrum Programmeurs f 26,75
 Machinetaal voor de ZX Spectrum f 34,75
 Werken met de ZX Microdrive f 33,50

BOEKEN voor de SPECTRUM

ZX Spectrum Hardware Boek f 26,50
 ZX Spectrum Machinetaalroutines f 33,75
 Financiële Programma's voor de ZX Spectrum f 25,75
 Toolkits en Enkele Spelen voor de ZX Spectrum .. f 25,75
 Van BASIC naar Machinetaal op de ZX Spectrum .. f 17,90
 Werkboek Machinetaal voor de ZX Spectrum f 37,90
 inclusief cassette met assembler

Exploring Artificial Intelligence on Spectrum . f 36,00
 Spectrum Assembly Language Course + tape f 69,00
 inclusief assembler op tape
 Complete Spectrum ROM disassembly f 49,00
 Spectrum Shadow ROM disassembly f 45,00

Advanced Spectrum Forth f 45,00
 Advanced Spectrum Machine Language f 36,00
 A Guide to playing the Hobbit f 20,00

BOEKEN voor de QL

Het Sinclair QL Handboek f 20,00
 Sinclair QL Leren programmeren f 24,75
 QL Advanced User Guide f 79,00
 Advanced QL Machine Code f 45,00
 Quick QL Machine Language f 39,00

*De M68000 Microprocessor f 49,50
 *68000 Assembly Language Programming 2nd edition f 69,00

ACTUELE EN NIEUW BINNENGEKOMEN SOFTWARE

SOFTWARE voor de SPECTRUM

SPECTRUM - Digital Drum System f 130
 Interface + tape
 Electro Kit voor Spectrum f 25
 Latin Kit voor Spectrum f 19

1942 f 36
 *Scooby Doo f 32
 Strike Force Harrier f 45
 *Konami Coin-Op Hits f 39
 oa Green Beret, Mikie, Hypersports
 Light Force f 36
 *Cobra - Stallone f 32
 Trap Door f 36
 *Unbelievable! Ultimate f 39
 oa JetPac, Tranzam
 The Great Escape f 36
 Glider Rider f 36
 *Gauntlet f 36
 *Academy (Tau Ceti 3) f 36
 Paperboy f 36
 Fat Worm Blows a Sparky f 39
 Asterix f 36
 *Starglider f 59
 Infiltrator f 39

SOFTWARE voor de SPECTRUM

They Sold a Million - 3 f 39
 Ghostbusters, Fighter Pilot,
 Rambo en Kung Fu Master.

*Fairlight II Trail of Darkness f 39
 *Aliens f 39
 *Olli and Lisa f 10
 *Avenger f 39
 *Firelord f 36
 *Storm f 10
 *Aftershock f 39
 *Contact Sam Cruise f 32
 Graphic Adventure Creator f 95

Colossus Chess 4.0 f 45
 *Colossus Bridge 4.0 (Acol) ... f 49

*TopGun (F14 Tomcat) f 32
 Elite f 65
 *Shaolins Road f 32
 *Footballer of the Year f 32
 *Konami's Golf f 32
 *Super Soccer f 32
 *Speed King 2 f 10
 *Hardball -honkbal f 36

SOFTWARE voor de SPECTRUM

DEVPAK editor/assembler f 59
 *Hisoft BASIC Compiler f 59
 Hisoft Pascal f 95
 Hisoft C Compiler f 95
 Arcade Creator f 65
 Beta BASIC 3.0 f 69
 Omnicalc II spreadsheet f 69
 Tasword III f 69
 Mini Office f 29
 database, spreadsheet, wordproc.

SOFTWARE voor de QL

QL Bridge Player II f 79
 QL Chess f 95
 The Lost Pharaoh f 65
 Karate QL f 65
 QL Matchpoint (3D Tennis) f 69
 Assembler Workbench f 110
 Hisoft Devpac QL f 165
 Metacomco Pascal QL f 375
 Hisoft Superbee f 129
 SuperBASIC Extension ROM
 Techni QL (2D CAD package) ... f 195

winkel open van woensdag t/m zaterdag tussen 11.00 en 17.00 (maandag/dinsdag gesloten) - alle prijzen inclusief BTW
 verzendkosten f 6,- per bestelling - vraag onze nieuwe WINTER 1986/87 CATALOGUS aan.

microcomputer tijdschriften boeken en software