

ARE ON "STATE STREET"

```

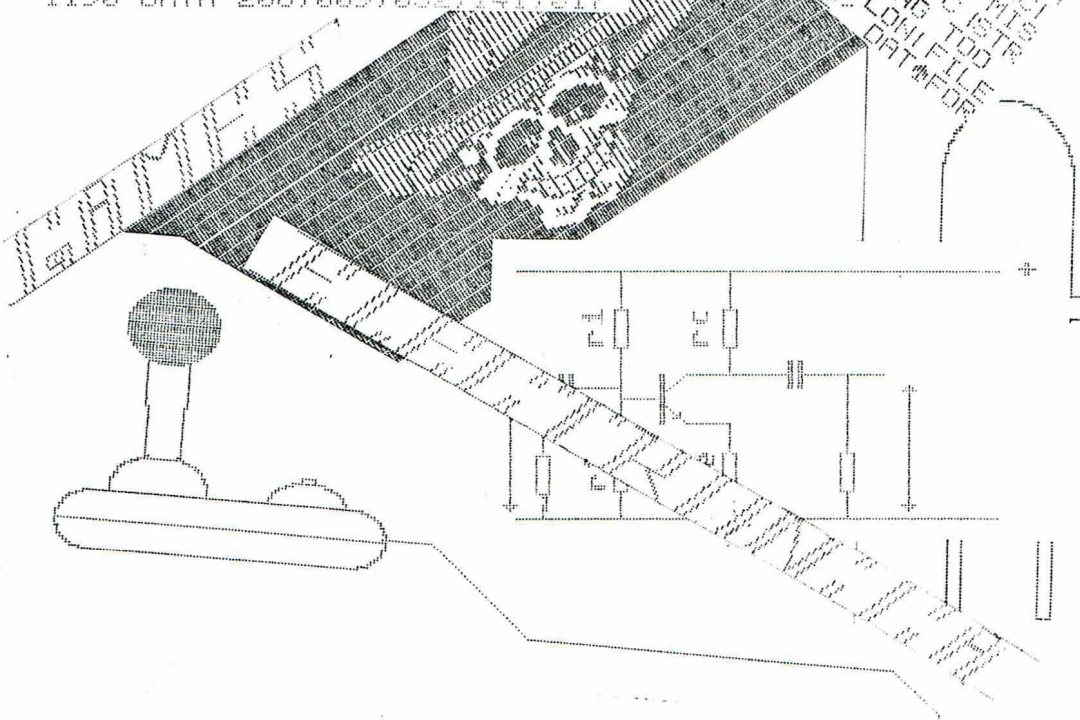
1000 0055,130:POKE56,159
1010 FORI= 40935 TO 40952
1020 READA:POKEI,A:NEXT
1030 PRINTCHR$(147),CHR$(144)
1040 PRINT"      CMP-SOFTWARE"
1050 PRINT"      SYS( 40935 )"
1060 PRINT"      HALF HIRES SCHEM"
1070 DATA 120,169,127,141,013
1080 DATA 220,169,001,141,025
1090 DATA 209,169

```

```

1100 DATA 003,169,159,141,021
1110 DATA 003,008,006,173,025
1120 DATA 208,141,025,208,041
1130 DATA 001,240,038,173,019
1140 DATA 208,201,154,240,034
1150 DATA 173,024,208,009,008
1160 DATA 141,024,208,173,017
1170 DATA 208,009,032,141,017

```



CMP SOFTWARE

FL.2.25 NO 2

INHOUD

SPEL TOP 5

- | | |
|-----------------------------------|---------------------|
| - MACHINETAAL cursus deel 2 | 1 SILENT SERVICE |
| - SOFTWARE Great am cross country | 2 ROCK AND WRESTLE |
| - BASIC | 3 WHO DARES WINS II |
| - SOFTWARE the dallas quest | 4 URIDIUM |
| - TEKST PRINTER | 5 ENIGMA FORCE |
| - ELECTRONICA | |
-

De REM 64 wordt samengesteld mbv een commodore 64.
De hierin beschreven en ontworpen programma's zijn alleen
voor deze computer geschreven.

De in dit nummer gepubliceerde programma's mogen niet
zonder toestemming voor commerciële doeleinden worden
gebruikt.

De samenstellers kunnen niet aansprakelijk worden gesteld
voor de eventuele schade die zou kunnen voortvloeien uit
een fout, die in deze uitgave zou kunnen voortkomen.

Zijn er op-en/of aanmerkingen laat het ons weten.

VOOR VRAAG EN/OF BESTELLING: TEL 03408-83882(jon)
85883(sjef)

NA ZESSEN BELLEN!!!!!!!!!!!!

HOE TE PROGRAMMEREN IN MACHINETAAL
OP DE COMMODORE 64 DEEL 2

Alvorens we gaan kijken naar de machine opdrachten van de 6510 microProcessor zullen we eerst de REGISTERS van de computer bekijken. Wat zijn nu de registers? Dit zijn speciale geheugen lokaties die de computer gebruikt om b.v data transport en alle andere machine instructies te kunnen uitvoeren.

De op dit moment voor ons belangrijkste registers zijn:

- A..... De ACCUMULATOR afgekort met ACCU of de letter A. Het A-register is het hoofd register en kan op elk moment een door ons gewenste waarde bevatten die in een byte mogelijk is. (van 0 tot 255).

- X..... Het X-register: Dit register kan net als de ACCU een byte bevatten en wordt meestal als teller in het programma gebruikt. In basic is dit vaak de FOR NEXT lus.

- Y..... Het Y-register dit heeft de zelfde functie als het X-register.

- P..... Het STATUS-register aangeduidt met de letter P. Het status-register wordt niet gebruikt om byte waarden in vast te leggen maar met behulp van de bit waarden van dit register kunnen we b.v vergelijkingen opstellen. Het gebruik van het status register zal bij de behandeling van de machine opdrachten vanzelf duidelijker worden. De opbouw van het status register is als volgt:

bitnr. 7 6 5 4 3 2 1 0
 N V - B D I Z C

C=carry, Z=zero, I=irq
D=decimaal, B=break, -=ongebruikt is altijd 1
V=overflow, N=negative.

De overige drie registers zijn de stack Pointer (S) dit wijst de eerst volgend vrije stack-adres aan, de Programcounters (PCH en PCL) deze counters wijzen naar het huidige programma adres (in basic termen de programma regel).

HOE TE PROGRAMEREN IN MACHINETAAL
OP DE COMMODORE 64 DEEL 2

Nu zullen we als eerste een korte beschrijving van de geheugen indeling van de '64 geven, later zullen we hier uitgebreider op ingaan omdat dit voor het vervaardigen van een machinetaal programma onmisbaar is. Waarbij we het nu over de lege geheugen plaatsen zullen hebben waar we ons machinetaal programma kunnen plaatsen.

bruikbaar zijn:

\$0002 (#2)	Is een onbenutte geheugenplaats nuttig als
\$00FB-00FE	Vier lege zero Pagina geheugens, b.v telgeheugen
\$0200-02FE (#704-766)	Hier zijn 63 lege geheugen plaatsen.
\$0330-03FB (#828-1019)	Indien men geen gebruik maakt van de cassette dan zijn hier 192 geheugen plaatsen vrij.
\$03FC-03FF (#1020-1023)	Vier lege geheugen plaatsen.
\$0800-9FFF (#2048-40959)	Indien er geen basic programma in het geheugen aanwezig is zijn er hier 38911 lege geheugen plaatsen (zie ook beginscherm van de computer).
Indien ons programma geheel in machinetaal is geschreven kunnen we ook de plaatsen \$A000 tot BFFF (40960-49151) gebruiken. We moeten dan de computer wel eerst duidelijk maken dat we het programma dat zich hier bevindt (het commodore basic) niet meer nodig hebben. Dit gebeurt door geheugen plaats #01 te veranderen. De originele waarde van deze geheugen plaats is #37 door deze nu te veranderen in #36 hebben we de beschikking over 8192 vrije geheugen plaatsen.	
\$C000-CFFF (#49152-53247)	Hier zijn 4096 vrije geheugen plaatsen.

De instructies zullen we nu op alfabetische volgorde behandelen zodat ze later gemakkelijk kunnen worden terug gezocht. Bij de uitleg van de opdrachten zullen er ook opdrachten gebruikt worden die nog niet behandeld zijn deze zullen toch zo goed mogelijk verklaart worden.

rem



O P G E L E T !!!!!!!!

COMPUTERCLUB OF NIET.

Bij voldoende belangstelling voor de rem 64 zijn we van Plan een computerclub op te richten waarbij de rem 64 het club blad zal zijn.. Het is de bedoeling dit blad minstens 4 maal Per jaar uit te brengen + nog enkele vlug schriften en natuurlijk bijeenkomsten. Via de rem 64 kan men dan adressen van mede 'terminal junk's' en natuurlijk programma's uitwisselen.

WE WANT YOU FOR OUR COMPUTER CLUB !!!!!

Ja ik heb belangstelling voor de club .

Alleen wanneer er voldoende belangstelling voor de oprichting van een club is zal deze opgericht worden. anders zullen we het voorlopig nog bij het blad moeten houden.

Laat het ons weten !!!!!



HOE TE PROGRAMMEREN IN MACHINETAAL
OP DE COMMODORE 64 DEEL 2

ADC ADD with Carry dit betekent als de computer deze opdracht
--- -- - tegen komt de waarde van het gespecificeerde
 geheugenadres of een vaste waarde bij de inhoud
 van de accu wordt opgeteld. Als nu de som groter
 wordt dan 255 dan wordt de carry flag van het
 status-register geset. Indien echter het resultaat
 gelijk is aan nul dan wordt de zero flag geset.
 een reken v.b: 128+128=256
 in machinetaal wordt dit nu:

```
CLC            reset de carry (C=0)
LDA##80        (A=128)
ADC##80        (tel er 128 bij op) + de
                  waarde van de carry flag
                  (hier 0 door clc)
```

De inhoud van de Accu-mulator is nu gelijk aan 00 daar
256 te groot is om met een byte voort te stellen.
Nu is echter de C-flag geset. Uit de binaire optelling
blijkt nu het beste waar dit voor gebruikt kan worden.

```
bitnr.76543210
10000000     Bij deze optelling blijkt nu duidelijk dat
+10000000     de som niet in een byte kan worden bevat
-----
100000000    daar de cmb64 slechts een acht bit's comp-
                  puter is. Om nu toch met grotere getallen te
                  kunnen rekenen moet u kijken of er
                  tijdens de optelling een carry plaats had. uit het
                  volgende voorbeeld blijkt het beste hoe u dat moet voor-
                  stellen.
```

```
CLC            reset carry
LDA##FF        A=255
ADC##03        tel er 3 bij op + de waarde
                  van de carry flag (die = 0)
```

De waarde van de accu is nu gelijk aan 02 de decimale notatie van de optelling
=258 (#0102) iederekeer als bij een optelling een carry plaatsvindt dan zal bij
een volgende optelling er een 1 bij de accu worden opgeteld.
Nu9 een reken voorbeeld: tel de getallen 1279 en 1106 bij elkaar op.
De hexadecimale notatie van 1279 is #04FF en van 1106 is dat #0452. De optelling
van 1279+1106=2385 dit is #0951 met een machinetaal Program-
ma gaat dit als volgt.

```
CLC            carry=0
LDA##FF        Lo-byte van 04FF (04=Hi, FF=Lo)
ADC##52        Tel daar de Lo-byte van 0452 bij op
STA#0340       Sta de accu inhoud op in geheugen
                  plaats #0340 (=832). + de waarde
                  van de carry flag (FF+52+0=51)

LDA##04        Hi-byte van 04FF
ADC##04        Tel daar de Hi-byte van 0452 bij op
                  + de waarde van de carry =1
                  (04+04+1=09).

STA#0341       Sta de accu inhoud op in geheugen
                  plaats 0341 (=833)
```

Het antwoord bevindt zich nu in de geheugen plaatsen 0340 en 0341 en wel als
volgt in 0340 bevindt zich de Lo-byte #51 en in 0341 bevindt zich de Hi-byte #09

HOE TE PROGRAMMEREN IN MACHINETAAL
OP DE COMMODORE 64 DEEL 2

Er zijn bij deze opdracht verder nog verschillende adresseer mogelijkheden om de op te tellen data getallen aan te duiden.

ADC##.. 'Onmiddellijk' De hex-waarde die achter deze opdracht in het geheugen staat wordt nu bij de accu-inhoud opgeteld.

ADC#.. 'Zero-Pagina' (dit zijn de eerste 256 byte's van het geheugen) De Hex-waarde achter deze opdracht bepaald het geheugenadres van de 'zero-Pagina' waarvan de inhoud bij de accu wordt opgeteld.
v.b.: ADC#02 De inhoud van geheugen \$02 wordt bij de accu-inhoud opgeteld.

ADC#...X 'Zero-Pagina' Het zelfde als boven maar nu wordt het geheugenadres bepaald door bij de hex-waarde achter de opdracht de x-waarde op te tellen. B.v.:
LDX##03 stel de waarde van x op 3
ADC#02,x tel bij de accu de waarde van geheugen 02+03=05 op.
Met behulp van deze adresseer methode kunnen 256 bytes worden aangesproken.

ADC#.... 'Absoluut' Met behulp van deze opdracht wordt de inhoud van een geheugen, dat aangeduid wordt met de twee byte's achter de opdracht, opgeteld bij de accu inhoud.
ADC#0400 tel de inhoud van \$0400 op bij de accu.

ADC#....X 'Absoluut' Het zelfde als boven nu wordt echter bij het samengestelde geheugenadres de x waarde opgeteld om het uiteindelijke geheugen waarvan de inhoud bij de accu wordt opgeteld aan te duiden.

ADC#....Y 'Absoluut' Nu met het Y-register.

ADC(\$...X) 'Zero-Pagina' Deze manier van adresseren heeft veel voordelen ten opzichte van de voorgaande. De HEX-waarde die achter de opdracht staat duidt nu niet ('direct') het geheugen adres aan maar het geheugen in de zero-pagina waarin zich de Lo-byte van het geheugen adres zich bevindt. In het opvolgende geheugen van de zero-pagina bevindt zich nu de Hi-byte.

B.v.:

1388 A9 00	LDA#\$00	Stel de accu inhoud op \$00.
138A 85 FD	STA#FD	Sta deze op in geheugen \$FD (=Lo-byte).
138C A9 08	LDA#\$08	Stel de accu inhoud op \$08.
138E 85 FE	STA#FE	Sta deze op in geheugen \$FE (=Hi-byte).
1390 A9 04	LDA#\$04	Stel de accu inhoud op \$04.
1392 8D 00 08	STA#0800	Sta deze op in geheugen \$0800 .
1395 18	CLC	Reset carry =0
1396 A2 00	LDX#\$00	Stel het x-register op \$00.
1398 A9 08	LDA#\$08	Stel de accu inhoud op \$08.
139A 61 FD	ADC(\$FD,X)	Tel er de inhoud van 0800+X bij op.
139C 8D 01 08	STA#0801	Sta het resultaat op in \$0801.
139F 60	RTS	Return.

HOE TE PROGRAMMEREN IN MACHINETAAL
OP DE COMMODORE 64 DEEL 2

In dit machinetaal programma wordt nu als eerste in de zero-Pagina adressen \$FD en \$FE de Lo- en de Hi-byte van de geheugen plaats waaruit de op te tellen waarde moet worden gehaald vastgelegd. In ons voorbeeld is dat geheugen \$0800 Dit bestaat uit \$00 dit is de Lo-byte deze wordt in geheugen \$FD geplaatst en \$08 deze wordt in geheugen \$FE geplaatst. Verder wordt in het geheugen \$0800 zelf de waarde \$04 geplaatst. Wanneer nu de computer de opdracht ADC(\$FD,X) tegen komt haalt de computer niet de op te tellen waarde uit \$FD maar de Lo-byte uit \$FD. In ons geval is dat \$00 tevens wordt uit de opvolgende byte \$FE de Hi-byte van ons geheugenadres \$0800 gehaald deze is dus \$08. De waarde die de computer bij de accu-inhoud optelt komt uit geheugen \$0800+X. Het X-register is (in ons geval) gelijk aan \$00 dus de waarde komt uit \$0800 en is \$04 dit hebben we er immers zelf net ingezet. Dit wordt nu bij de accu-inhoud \$08 opgeteld en het resultaat wordt in geheugen \$0801 opgeslagen.

ADC(\$..),Y Deze opdracht is hetzelfde nu echter wordt het Y-register gebruikt.

In tabel vorm :

Opdracht:	Hex notatie	Dec notatie	geheugen gebruik.
ADC\$ Zero Pagina	\$65	101	2 -byte's
ADC\$ Zero Pagina,X	\$75	117	2 -byte's
ADC(\$Zero Pagina,X)	\$61	97	2 -byte's
ADC(\$Zero Pagina),Y	\$71	113	2 -byte's
ADC##Onmiddellijk	\$69	105	2 -byte's
ADC\$ Absoluut	\$6D	109	3 -byte's
ADC\$ Absoluut,X	\$7D	125	3 -byte's
ADC\$ Absoluut,Y	\$79	121	3 -byte's

Verder is het van belang te weten dat de ADC opdracht de volgende de bit's van het status register beïnvloed.
bit 7 de N-flag, bit 6 de V-flag, bit 1 de Z-flag en bit 0 de C-flag.

Statusregister	7	6	5	4	3	2	1	0
	N	V	-	B	D	I	Z	C
	↑	↑					↑	↑

In deze aflevering hebben we door de complexiteit van de opdracht slechts een opdracht kunnen behandelen volgende keer meer voor vragen staan we altijd open tel 03408-85803(sjef), 03408-83002(jon).

THE GREAT AMERICAN CROSS-COUNTRY ROAD RACE™

THE GREAT AMERICAN CROSS-COUNTRY ROAD RACE

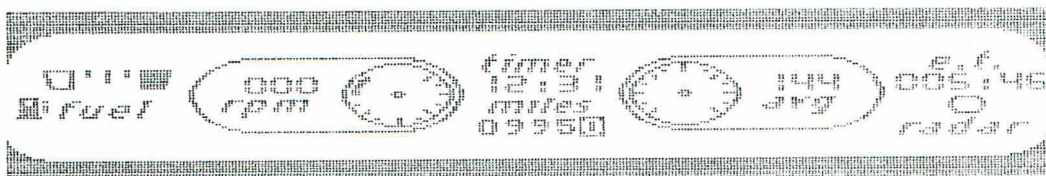
Deze omvangrijke titel hoort bij een auto simulator.
Dit keer eens niet autoracen maar rally rijden.
Het begin van het spel is vrij stoffig dwz een aantal namen verschiint en ver-
verduijnt met de komst van een stofwolk die veroorzaakt wordt door een rally-
auto.

De functietoets F1 laat de Kaart van Amerika zien, dan kan met de joystick
(Port 1) besloten worden naar welke Plaats men rijdt.
Druk dan op 'fire' en het spel start.
Het is verstandiger om eerst op F3 te drukken.
Daarmee verkrijgt men de mogelijkheid om een Klassement van disk in te laden.
Er zijn 8mogelijkheden.
Dit laden hoeft niet!
Dan is er de keuze uit 4 verschillende routes en wedstrijden.

- 1 Los Angeles to New York
- 2 Seattle to Miami
- 3 San Fransisco to Washinton
- 4 U.S. tour

De eerste drie zijn routes van de west- naar de oost kust.
De laatste is een rit van Seattle naar Seattle via de oostkust.
Op de Kaart staat ijzel (sterretjes) en regenbuien voor gladheid.
Dit wordt in het spel vertolkt door Plekken op de weg waardoor, als je erdoor
rijdt, naar de kant glijd.

Het dashboard van je voertuig is vrij uitgebreid vergeleken met andere autospel
letjes.
Op onderstaand Plaatje, van het dashboard, worden alle klokjes en meters ver-
klaard.



- 1 de Benzine meter
- 2 de toerenteller (van de motor)
- 3 de Timer ; deze houdt de tijd bij waarop je binnen moet zijn.
Hij begint bij 30 seconden waarschuwend te tikken om je te laten merken dat je op moet schieten.
- 4 de Afstandmeter deze geeft het aantal mijlen aan dat je nog af moet leggen.
- 5 de Toerenteller net als de afstandmeter in mijlen.
- 6 de waarschuwings lamp

Als die lamp begint te knipperen wordt je door de American Highway Patrol achterna gezeten.

Dan is het zaak zo hard mogelijk te gaan om hem te ontlopen.

Het sturen gaat dan wel gemakkelijker omdat alle andere auto's verdwenen zijn.

Bots je tegen een -Politie-auto op dan kost je dit enige seconden en moet je je snelheid weer helemaal opbouwen.

De besturing van je auto is wat uitgebreider dan bij andere race spelletjes.

Bij dit spel behoor je namelijk te schakelen.

Dit schakelen maakt het geheel wel veel realistischer.

De snelheid voer je op door op 'fire' te drukken.

wanneer de motor genoeg toeren maakt, druk dan de knuppel naar voren.

Dan schakel je naar een hogere versnelling.

Dit kan door blijven gaan tot en met 240 mph, dat is de hoogst haalbare snelheid.

Laat de motor nooit te lang te hard lopen durf te veel toeren maken, schakel op tijd naar een hogere versnelling.

Doe je dit niet dan zal de motor exploderen waarna je mag pompen 'fire' naar een benzine station.

Dit geldt ook als je benzine op is.

SAN FRANCISCO TO WASHINGTON

	NAME	DATE	MPH	TIME
1	FAST GAS GARRY	04/27	145	021:11
2	LEADFOOT LAURETTE	04/13	140	021:56
3	QUICK START K.C.	04/30	135	022:45
4	DARING DAN	04/20	130	023:38
5	JUMP START JOHN	04/30	124	024:35
6	MOVE OVER MARY	04/09	120	025:46
7	CAREFUL CAROLINE	04/14	114	026:43
8	JERSEY JESSICA	04/06	109	027:56
9	JET ENGINE JIM	04/18	105	029:15
10	KEEP AHEAD KEVIN	04/13	100	030:43

LOS ANGELES TO NEW YORK

	NAME	DATE	MPH	TIME
1	ALL AMERICAN AL	04/06	145	019:18
2	EASTSIDE EDDIE	04/13	140	019:59
3	QUICK START K.C.	04/15	135	020:43
4	DARING DAN	04/19	130	021:31
5	JUMP START JOHN	04/30	125	022:23
6	MOVE OVER MARY	04/12	120	023:19
7	CAREFUL CAROLINE	04/21	115	024:20
8	JERSEY JESSICA	04/04	110	025:26
9	JET ENGINE JIM	04/10	105	026:39
10	KEEP AHEAD KEVIN	04/22	100	027:59

Na elke gereden etape wordt je, in de loopband, verteld op welke plaats je staat en achter wie.

Dit is een indicatie of je de volgende etape voorzichtig kan rijden of dat je haast moet gaan maken om nog een respectabele plaats te kunnen bereiken.

Na afloop kan het klassement - hierboven een voorbeeld - gesaved worden.

Dit geldt alleen voor disk.

Er is een keuze uit 8 verschillende klassementen.

Bij het inladen van het klassement moet weer hetzelfde nummer ingetypt worden.

BEGINNERS ALLPURPOSE SYMBOLIC INSTRUCTION CODE 2.

B A S I C

Als men vanuit een basic Programma een Programma (of meestal data) in wil laden en men geeft in het Programma simpel weg de opdracht LOAD"Programmanaam",8,1 dan komt men tot de ontdekking dat het Programma zelf na laden weer helemaal opnieuw begint of nog erger als men aan het begin van het Programma in de eerste regel het load commando geeft dan blijft de computer laden.

dit komt doordat iedere keer als de computer klaar is met de load opdracht hij naar het begin van het Programma terug keert en dan weer de LOAD opdracht tegen komt enz.

Met behulp van onderstaande miniatuurtjes wordt dit probleem omzeilt.

1ste

```
100 IF FLAG = 1 THEN 150 :Manner er nu een fout melding op-
110 A$="PROGRAMMA NAAM":FLAG = 1 :treedt wordt het basic Programma
120 PRINT"LOAD"CHR$(34);A$;CHR$(34)",8,1" :zonder broken en wordt de fout afge-
130 FOR X=0 TO 2:PRINTCHR$(145):;NEXT :drukt.
140 POKE631,13:POKE198,1
150 REM VERVOLG PROGRAMMA
```

2 de

```
100 IF FLAG = 1 THEN 130
110 A$="PROGRAMMA NAAM":FLAG = 1
120 LOADA$,8,1
130 REM VERVOLG PROGRAMMA
```

FOUT UITLEES ROUTINE:

```
100 OPEN1,1,15 :Nadat u dit Programma heeft gerunt bevat A het fout nr
110 INPUT#1,A,B$,C,D :;0=geen fout ,B$ de fout omschrijving b.v.'FILE NOT
120 PRINTA;B$;C;D :FOUND', C het spoor nr en D het sector nr dit kan van
130 CLOSE1 :belang zijn bij bv een 'READ ERROR'.
:voor de uitgebreide lijst van fout meldingen dient
men de handleiding van de disk drive te raadplegen.
```

Intypers bij het Programma voor een half hires scherm zijn de onderstreepte data getallen de kleuren van beeldrand en scherm.Het hires beeld moet op 8192 beginnen.

```
1000 POKE55,130:POKE56,159
1010 FORI= 40835 TO 40952
1020 READA:POKEI,A:NEXT
1030 PRINTCHR$(147),CHR$(144)
1040 PRINT" CMP-SOFTWARE"
1050 PRINT"HALF HIRES SCHERM"
1060 PRINT" SYS( 40835 )"
1070 DATA 120,169,127,141,013
1080 DATA 220,169,001,141,026
1090 DATA 208,169,000,141,018
1100 DATA 208,169,027,141,017
1110 DATA 208,169,164,141,020
1120 DATA 003,169,159,141,021
1130 DATA 003,098,096,173,025
1140 DATA 208,141,025,208,041
1150 DATA 001,240,038,173,018
1160 DATA 208,201,154,240,034
1170 DATA 173,024,208,009,008
1180 DATA 141,024,208,173,017
1190 DATA 208,009,032,141,017
1200 DATA 208,169,012,162,006
1210 DATA 141,032,208,142,033
1220 DATA 208,169,154,141,018
1230 DATA 208,076,049,234,173
1240 DATA 017,208,041,223,141
1250 DATA 017,208,173,024,208
1260 DATA 041,247,141,024,208
1270 DATA 169,006,162,004,141
1280 DATA 032,208,142,033,208
1290 DATA 169,000,141,018,208
1300 DATA 076,188,254
1000 FOR I= 724 TO 760
1010 READA:POKEI,A:NEXT
1020 PRINTCHR$(147),CHR$(144)
1025 PRINT" CMP.....SAVER"
1030 PRINT:PRINT:PRINT:PRINT
1040 INPUT"TAPE OR DISK :";D$
1050 IFD$="D"THENEND=8
1060 IFD$="T"THENEND=1
1070 INPUT"FILENAME :";F$
1080 L=LEN(F$):IFL>16THEN1070
1090 IFL<1THEN1070
1100 FORK=1TOL:C=ASC(MID$(F$,K,1))
1110 POKE705+K,C:NEXTK:PRINT
1120 INPUT"START (DEC):";SA
1130 INPUT"END (DEC):";EA
1140 HS=INT(SA/256):LS=SA-HS*256
1150 HE=INT(EA/256):LE=EA-HE*256
1160 POKE743,LS:POKE747,HS
1170 POKE751,LE:POKE753,HE
1180 POKE704,D:POKE705,L
1190 SYS724:END
1200 DATA 173,192,002,168,170
1210 DATA 032,186,255,173,193
1220 DATA 002,162,194,160,002
1230 DATA 032,189,255,169,000
1240 DATA 133,251,169,000,133
1250 DATA 252,162,000,160,000
1260 DATA 169,251,032,216,255
1270 DATA 088,096
```

THE DALLAS QUEST

De Dallas Quest verschilt in veel opzichten met andere adventures.
Het is namelijk niet een spel waarbij je niet zonder een Plattegrond Kan.
Bij de Dallas Quest is het meer een kwestie van Goed nadenken.
Je zit in de moeilijkheden en wat doen we er aan.
Qua woorden is de Dallas Quest zeer uitgebreid.
De computer begrijpt al gauw wat er bedoelt wordt.
Er is ook een afkortings mogelijkheid ingebouwd.
Flashlight - zaklantaarn - kan afgekort worden tot de eerste vier letters, dus
flas.

Het spel:

Bij de start van het spel bevindt men zich in het huis - villa - van J.R.
De eerste, uit de galerij de Dallas figuren, die je tegenkomt is Sue Ellen.
Wat je ook probeert daar valt niets mee uit te halen.
Van haar krijg je de opdracht, je bent detective, een kaart op te sporen.
Maar je kan het huis niet uit omdat J.R. je daar gevangen houdt (Probeer de
voordeur maar eens).
Het getal onder de linker boortoren houd bij hoeveel keer je een commando
ingetypt hebt.
Het getal onder de rechter boortoren geeft aan hoeveel clue's - help - je nog
op kan vragen.
Het gaat te ver om te beschrijven wat je, stap voor stap, moet doen daarom
wordt deze beschrijving beperkt tot wat tips.

Het Huis

- een nachtuil + een zonnebril = een daguil
- uilen lusten graag ratten
- graven kan nuttig zijn
- onrustige koeien houden van muziek
- tabak -Pouch- vinden apen lekker

Als je zover bent bezit je een code.
Volg die code op als je door het Korenveld loopt en je vindt Ray Krebs.
Om nu aan J.R. te kunnen ontsnappen zijn er de volgende tips:

- Ray is een Ewing
- draag in een vliegtuig altijd een parachute
- een knapzak is altijd handig

De jungle

In het oerwoud moet je zorgen dat je de weg niet kwijt raakt; blijf dus op het
pad; duz twee maal zuid.

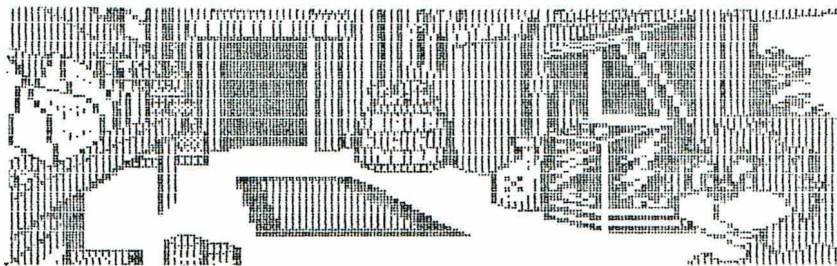
- PapaSaaiën Kunnen Praten
- slangen kunnen NIET tegen Kietelen

De rivier

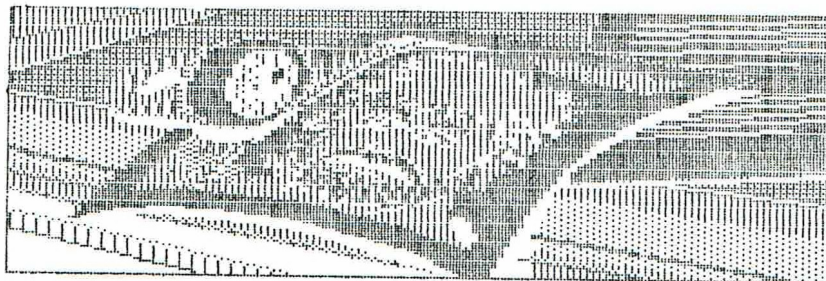
- alweer kan een stukje muziek helpen

De handelsPost

De handelsPost is van chugalus



Hiervan bezit je een foto; verlies die niet!
Evenals de foto is de ring ook van Grote waarde.
Beide voorwerpen zijn je in bruikleen gegeven door Sue Ellen.

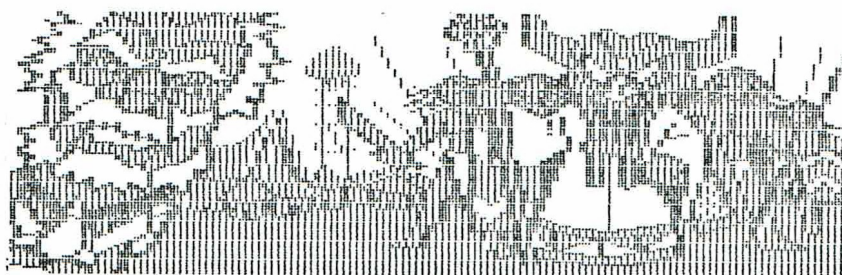


Op de wankelende ladder kan je maar een ding tegelijk dragen; dus eerst de zak-
lantaren - neerleggen (brandende) - terug naar de handelsPost en nog een keer
de ladder af met de knapzak.

>Read sign and go west<

Het inboorlingen dorp

Het opperhoofd is niet onder de indruk van je dans capaciteiten.
Verzoen je met hem door een opdracht te vervullen.



Keuze:

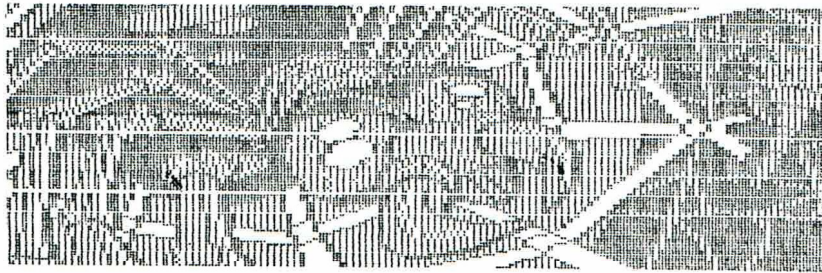
	richting	dier	hulpmiddel
MANE STREET	- noord	- leeuw	- spiegel
DUMBO'S NEST	- zuid	- olifant	- ring
NEST EGG	- west	- gier	- kokosnoot

SOFTWARE*SOFTWARE*SOFTWARE*SOFTWARE*SOFTWARE*SOFTWARE*SOFTWARE*SOFTWARE*SOFTWARE

Het is de bedoeling dat je weet wat er in Mane street en bij Dumbo's nest moet gebeuren.

Mant die kennis heb je nodig als je terug in het dorp bent.

De spin



Hier voldoet maar 1 tip en dat is verwarm de eieren.

Gefeliciteerd je hebt de opdracht vervuld.

Natuurlijk zijn er nog genoeg dingen die hierin niet beschreven zijn omdat je zelf ook nog wat mag oplossen.

Nu volgt een lijst van werkwoorden die je nodig hebt om dit adventure tot een goed einde te brengen

Allerlaatste tip : vertrouw J.R. NOOOOOOOOOOOOOOIT

look examine observe go enter get take drop put clue read climb inventory(I)
quit play blow wear ride swim open(O) close(C) wait shoot fire eat tast chew
feed give offer kill dig remove save load bribe pay hire fly jump row paddle
tick help talk speak pull draw light-on show wave heat warm use shit+fuc Yes
no north south east west

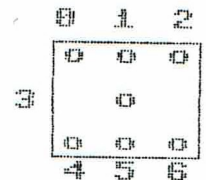
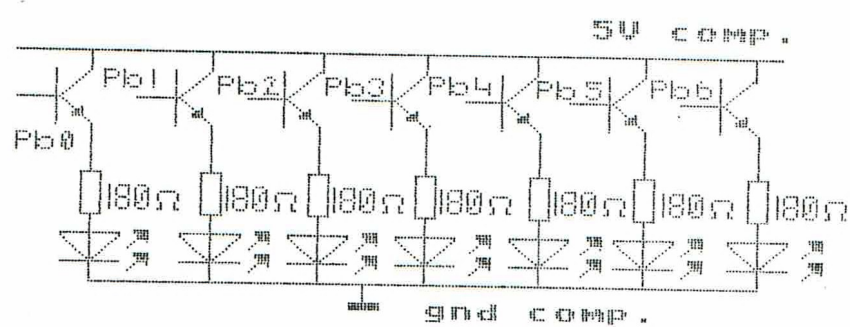
CMP-SOFTWARE TEXT PRINTER

```

100 PRINTCHR$(147),CHR$(144)
110 GOSUB410:PRINT"CMP-SOFT CHARACTER PRINTING":GOSUB410:GOSUB430
120 GOSUB430:PRINT"LENGTE.....":
130 GOSUB420:PRINT"BREEDTE.....":
140 GOSUB420:PRINT"CHARACTER.....":
150 GOSUB420:PRINT"GEEF DE TE PRINTEN TEXT...":
160 FORX=0TO12:PRINTCHR$(145);:NEXTX
170 GOSUB440:INPUTLL:GOSUB420
180 GOSUB440:INPUTBT:GOSUB420
190 GOSUB440:INPUTCH#:GOSUB420
200 GOSUB440:INPUTTPT#:GOSUB420
210 IFTPT#=""THENTPT#=""
220 FORX=0TOLL:L1#=L1#+CH#:L2#=L2#+"":NEXTX
230 L=LEN(TPT#)
240 FORLL=LTO1STEP-1:L#=MID$(TPT#,LL,1)
250 GOSUB260:NEXTLL:GOSUB450
260 W=ASC(L#):IFW>31ANDW<64THEN280
270 W=W-64:
280 POKE56334,0
290 POKE1,PEEK(1)AND251:FORN=0TO7:Q=W*8+N
300 BIT(N)=PEEK(53248+Q):NEXTN
310 POKE1,PEEK(1)OR4:POKE56334,1
320 FORK=7TO0STEP-1:FORY=0TO7
330 B=BIT(Y)AND(2^K):
340 IFB=2^K THENBIT$(K)=BIT$(K)+L1#
350 IFB<>2^KTHENBIT$(K)=BIT$(K)+L2#
360 NEXTY:NEXTK
370 OPEN4,4:CMD4:FORX=0TO7
380 FORKK=0TOBT:PRINTBIT$(X):NEXTKK
390 BIT$(X)="" :NEXTX:PRINT#4:CLOSE4
400 RETURN
410 FORC=0TO39:PRINTCHR$(100);:NEXTC:RETURN
420 PRINTCHR$(17);
430 PRINTCHR$(17):RETURN
440 FORX=0TO26:PRINTCHR$(29);:NEXT:RETURN
450 PRINTCHR$(147):GOSUB420:GOSUB420:GOSUB420
460 GOSUB410:PRINTCHR$(5)"STOP J/N.....":GOSUB410
470 FORX=0TO1:PRINTCHR$(145);:NEXTX:GOSUB440
480 PRINTCHR$(144);:INPUTJ#:IFJ#="N"THENCLOSE4:RUN
490 PRINTCHR$(147):END

```

Door het gebrek aan tijd schiet de electronica in dit nummer er wat bij in, maar we beloven dat dit in de volgende nummers weer goed gemaakt wordt.



de vorm waarin de ledjes moeten komen.

Het is een zeer eenvoudige schakeling. Eigenlijk is het 7 maal dezelfde schakeling.

De bedoeling is dat al die 7 schakelingetjes, die aan elkaar zitten, op een print zet.

De ledjes komen in een omgekeerde 'H' vorm.

De bijgegeven programmaatjes zullen de ledjes aansturen.

Knoop de ingangen van de schakelingetjes aan de userpoort.

Zorg wel dat je de goede pinnen van de userpoort neemt anders werkt hij niet of werkt hij nooit meer.

succes!!!!!!!

DOBBELSTEEN

```
10 rem de dobbelsteen
20 r=1+int((rnd(1)*6)
25 poke56579,127
30 ifr=1thenpoke56577,8
40 ifr=2thenpoke56577,20
50 ifr=3thenpoke56577,137
60 ifr=4thenpoke56577,149
70 ifr=5thenpoke56577,157
80 ifr=6thenpoke56577,183
90 geta$:ifa$=""then90
100 goto10
```

LOOPLICHT

```
10 rem looplicht
20 A=1
30 poke56579,127
40 gosub100
50 A=A*2
60 goto40
100 poke56577,A
110 forx=0to100:next
120 return
```

BINAIR

```
10 rem binair
20 poke56579,127
30 fori=1to255
40 poke56577,i
50 forx=0to100:nextx,i
```

CORRECTIE!!!!

bij LOOPLICHT moet onderstaande regel tussengevoegt worden

```
55 ifa>255thena=0
```