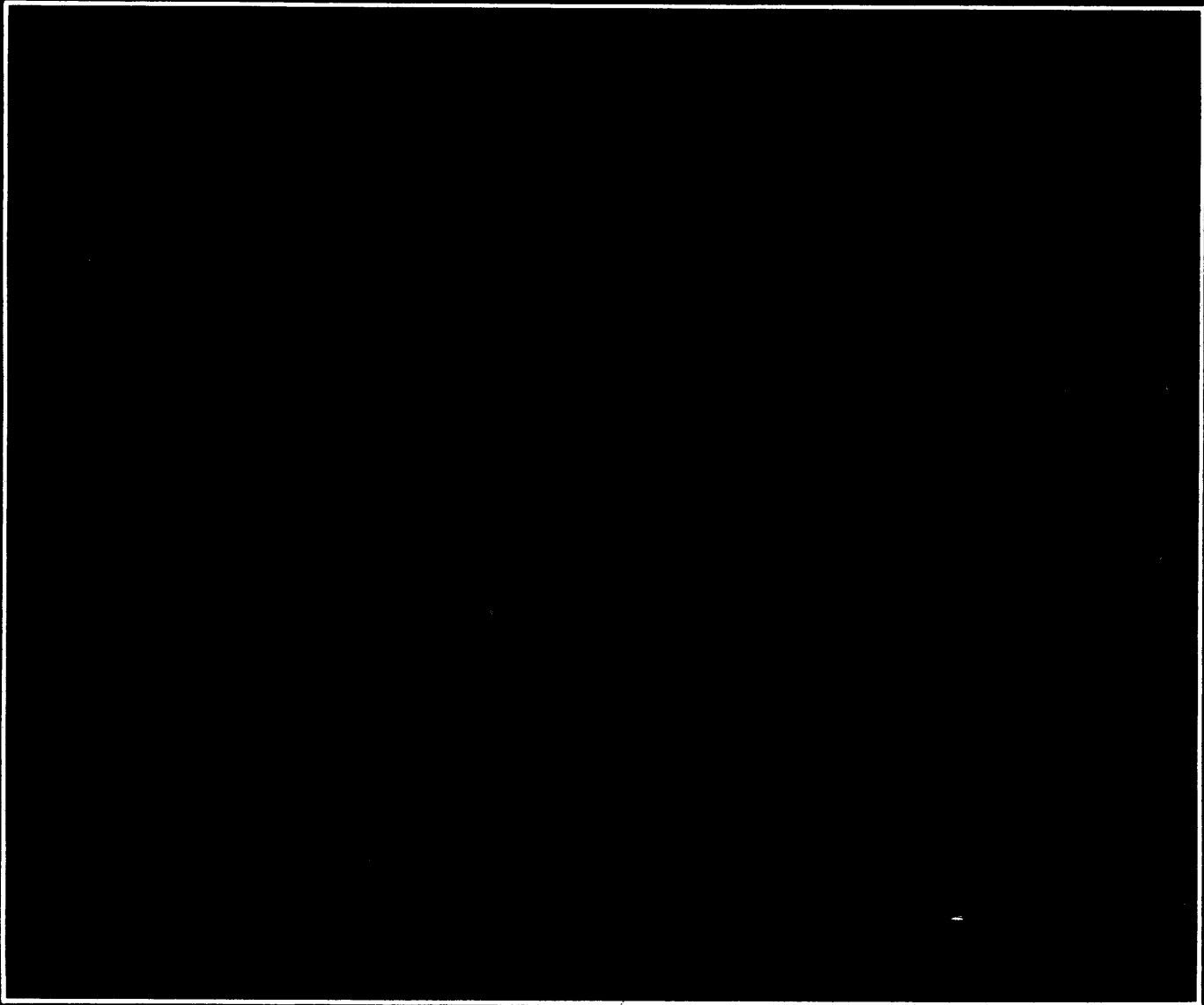


tweemaandelijks tijdschrift juli - augustus 1983



een uitgave van dainamic v.z.w.
verantw. uitgever w. hermans, heide 4 - 3171 westmeerbeek

COLOFON

DAInamic verschijnt tweemaandelijks.
Abonnementprijs is inbegrepen in de jaarlijkse
contributie .
Bij toetreding worden de verschenen nummers van de
jaargang toegezonden.

DAInamic redactie :

Dirk Bonné
Freddy De Raedt
Wilfried Hermans
René Rens
Jos Schepens
Roger Theeuws
Bruno Van Rompaey
Jef Verwimp

Vormgeving : Ludo Van Mechelen.

U wordt lid door storting van de contributie op het
rekeningnr. **230-0045353-74** van de **Generale
Bankmaatschappij, Leuven**, via bankinstelling of
postgiro
Het abonnement loopt van januari tot december.

DAInamic verschijnt de pare maanden.
Bijdragen zijn steeds welkom.

CORRESPONDENTIE ADRESSEN.

Redactie en software bibliotheek

Wilfried Hermans
Heide 4
B 3171 Westmeerbeek
België
tel. : 016/69.86.23

Kredietbank Westmeerbeek
nr. 406-3016141-33
BTW : 420.840.834

Lidgelden

Bruno Van Rompaey
Bovenbosstraat 4
B 3044 Haasrode
België
tel. : 016/46.10.85

Generale Bankmaatschappij Leuven
nr. 230-0045353-74

Inzendingen : Games & Strategy

Frank Druijff
's Gravendijkwal 5A
NL 3021 EA Rotterdam
Nederland
tel. : 010/25.42.75

DAInamic

PERSONAL COMPUTER USERS CLUB

4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

belangrijke ASCII-waarden in DAInpc

functie/symbool	HEX	DEC
back-space	8	8
TAB	9	9
linefeed	A	10
clear screen	C	12
CURSOR UP	10	16
CURSOR DOWN	11	17
CURSOR LEFT	12	18
CURSOR RIGHT	13	19
space-bar	20	32
Ø	30	48
A	41	65
a	61	97
pijltje rechts	89	137
pijltje links	88	136
pijltje boven	5E	94
pijltje onder	8C	140
volle blok	FF	255
verticale lijn	A	10
horizontale lijn	B	11
6 hor. lijnen	1D	29

ASCII - HEX - ASCII CONVERSION TABLE

MSD	0	1	2	3	4	5	6	7	
LSD	000	001	010	011	100	101	110	111	
0	0000	NUL	DLE	SP	0	●	P	ˆ	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	§	4	D	T	d	t
5	0101	ENC	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	VS	/	?	O	←	o	DEL

Beste leden,

Het zal zowat drie jaar geleden zijn dat de eerste publicatie van DAI namic de deur uitging. Een sober gestencild papiertje, verspreid via de firma DAI, kondigde aan dat er een gebruikersclub gesticht was rond de DAI personal computer. Het gebrek aan informatie omtrent dit revolutionaire toestel was mede de aanleiding geweest om contact te zoeken met medegebruikers. Het was de bedoeling mensen uit de regio te vinden om samen onze zo beruchte computer te leren gebruiken. De enorme bijval uit het buitenland, vooral Nederland tijdens de eerste maanden, was voor onze kern dan ook een enorme en prettige verrassing. De gebruikers van het eerste uur zullen zich zeker nog herinneren welke moeilijkheden er moesten worden overwonnen om een DAI computer in zijn bezit te krijgen. Velen van ons moesten het dan nog stellen met een 8K-versie zwart-wit, zonder sound. Rond die tijd was het voor DAI een enorme tegenvaller dat de toestellen voor de TELEAC-cursus niet tijdig konden geleverd worden, een enorme kans op ruime promotie ging hierdoor verloren. We mogen stellen dat alle toestellen die toen in productie genomen werden reeds lang op voorhand verkocht waren en de TELEAC-affaire was vlug vergeten. Helaas was de armslag van DAI te klein en kon de productie niet opgevoerd worden om aan de enorme aanvraag te voldoen. In Frankrijk, Engeland, Duitsland en Italië was er zeer veel belangstelling en DAI namic werd overspoeld met aanvragen in de verschillende landstalen. Het lag dan ook voor de hand om af te stappen van het uni-Nederlands en zodoende zijn de verschillende vertaaldiensten op gang gekomen. Erg spijtig en onzeker waren de gebeurtenissen omtrent het faillissement van de toenmalige firma. We zijn dan ook tevreden dat er uiteindelijk een gezonde overname gekomen is: INDATA was de naam. Nieuwe mensen, nieuwe politiek: het is ondertussen duidelijk geworden dat er meer belangstelling was voor de thuismarkt. Deze opstelling heeft er voor gezorgd dat het aantal Belgische leden reeds meer dan 450 bedraagt! In deze korte historiek mogen we een paar namen zeker niet vergeten: JC Camby, die als een ware diplomaat maar steeds weer de ongeduldige kopers en wachtenden moest te woord staan, Frank Druijff, die spoedig de Belgische kern kwam vervoegen, Freddy De Raedt, die zorgde voor programma's als FGT en Assembler en die ook de mensen die meer wilde weten over machinetaal te woord kon staan, Hans Wegman zette zijn mijlpaal in het DAI gebeuren met de ontwikkeling van MDCR, Jan Boerrigter die met zijn collega's er voor zorgde dat de hardware-geheimen van DAI voor ieder ontbloot werden en dan verder ging met zijn Firmware-manual. Bruno Van Rompaey nam de onderwijsmensen op sleep en schudde diDAIsoft uit zijn mouw... Vele naaste en verre medewerkers en correspondenten moeten we ongenoemd laten wegens plaatsgebrek. Allen samen hebben zij er voor gezorgd dat DAIpc zijn plaats op de turbulente computermarkt heeft weten te behouden en nog een gezonde toekomst voor zich heeft. We danken U voor de vele plezierige contacten,

tot volgende keer,

Dear members,

DAI namic is 3 years young, this short historical review is too hard (and too long) to translate, you will find a translation in one of the next issues. To celebrate our club-birthday, we have a special software-offer, see the card in this issue.

yours sincerely,

Wilfried Hermans

215	Remark	Redactie
216	Inhoudstafel-contents	
217	Software contributions	F.Druijff
218	Videotex in Belgium	PUB
219	16 couleurs charcateres	A.Mariatte
220	cassette tape lister	D.Assink
224	Programmeer technieken	F.druijff
228	The EDITOR story	J.Boerrigter
231	corrections firmware manual	J.Boerrigter
232	8080 cassette routines SDK-85	J.van Ool
234	bootstrap for screen files	W.Dewinter
244	Programmes mathematiques	F.Duluins
247	Programgenerator	N.Looije
248	Shadeshape	H.di Ciris
249	INDATA news	INDATA
250	AZERTY user	J.Schepens
252	comparison between BASIC & mlp	G.Uliana
253	RUN linenummer BASIC V1.0 (t)	C.Dufour
254	new PADDLE-check routine	C.Dufour
256	screen-buffer buffer screen demo	C.De Bont
257	ON ERROR GOTO demo	C.De Bont
258	DAI VIDEO SCREEN RAM (translation)	UK
264	programming techniques (t)	UK
267	CONVERSION APPLE-ATARI-DAI (t)	UK
269	TV-tennis REM's (t)	UK
270	cursus microprocessoren part 1	A.Beuckelaers
276	Shape-design	W.Hermans
278	Copieervariaties	B.Vingerling

DAInamic subscription rates :

Benelux	: 900 Bfr
Europe	: 1000 Bfr
Outside Europe	: 1400 Bfr
(Air Mail)	

pay to : Dainamic SUBSCRIPTIONS

B.Van rompaey
Bovenbosstraat 4
3044 HAASRODE-BELGIUM

* by check or
* on Bancaccount nr 230-0045353-74
of Generale Bank Leuven c/o DAInamic

Software contributions

S O F T W A R E - < CONTRIBUTIONS - INZENDINGEN - BEITRAGEN > - S O F T W A R E

Its holyday-time. We want to get some rest. Please have some patience if you are waiting for some answer or reply.

Some questions asked by more than one will be answered right now.

1- Q: Is it possible to protect a program against unauthorised copying ?

A: No. It is always possible to copy a program but we can make it rather difficult to do so. If you have a program of which you think it's good enough to be used by the club but would need some protection you can send it to us with this remark. If we agree, we will take care for the protection. The methods of protection we do use cannot be explained because someone who wants to break the protection will then be provided by just the facts he wants to know for this breaking. However there are some protective measures you can do yourself; these are rather simple but gives the person who wants to change or copy the program more work to do. The measures will be discussed in one of the following issue's of DAIInamic.

2- Q: Can I send my program for just comment. So I don't want give permission to use it in any way.

A: Yes. But if the program is good we will try to change your mind.

3- Q: Can I see a program before buying it

A: Yes. On the meetings where DAIInamic is present with software you can see the new software and ask questions about it

4- Q: Can I have a program sometime before deciding to buy.

A: No

5- Q: Do I get a garantee if I send you software that you don't give it away?

A: The software that we receive is only distributed among those members of which you can expect answer. We automaticly assume that we can use the software for our library. If you tell explicitly it is not to be used we obey. You can even ask for no distribution at all; in that case your software will stay with the person you send it to.

6- Q: Can I order software on floppy?

A: On this moment no. The floppy is still so uncommon that it would cost much to do so. Maybe in near future as many people have a floppydrive.

7- Q: How long will it take to get an answer?

A: Normaly (no holydays) you will have a letter notifying you that we did receive your software within a week. In that letter will be information on the person answering you. I try the completely handle it within two month, but that didn't work out always.

De staatssekretaris van PTT, mevrouw Paula d'Hondt heeft half april de modem vrijgegeven om zich niet tegen de technische evolutie op dat gebied te verzetten. Daarmee heeft zij dan ook een einde gemaakt aan een toestand, die door de gebruikers van het videotex-systeem als een klucht werd ervaren. De modem van de RTT huurde iedereen wel, maar gebruiken was een tweede. De grootste rem op de ontwikkeling van viewdata in België is daarmee in ieder geval weggenomen. Blijft alleen nog het relatief kleine probleem van de telefoonkosten.

"Wat is interactieve videotex?" vraagt men zich af in een publicatie van de RTT. *"Videotex is een communicatiesysteem dat o.a. toelaat informatie, opgeslagen in een komputergeheugen te verspreiden en op te vragen. Daartoe heeft de gebruiker een telefoon nodig evenals een terminal bestaande uit een beeldscherm en een klavier. (...) Interactieve videotex is een systeem met individuele tweewegtransmissie dat aldus een dialoog tussen de gebruiker en komputertoe laat. De opslagmogelijkheden van het hele systeem zijn slechts beperkt door de geïnstalleerde stockeringscapaciteit. De bediening is zeer eenvoudig en door enkele toetsen in te drukken krijgt de gebruiker alle gekozen informatie bijna onmiddellijk op het scherm. Dank zij de interactie zijn ook bewerkingen en transacties mogelijk."* En: *"Voor het bedrijfsleven. Naast een informatiebron is videotex een goed communicatiesysteem om relaties met geografisch gespreide filialen, verdelers, kantoren*

of personeelskernen te onderhouden. Een snelle verspreiding van veelgebruikte en vaak gewijzigde gegevens naar vele medewerkers is dan mogelijk. Ook intern voor een groot bedrijf met vele afdelingen en diensten kan videotex de communicatieproblemen oplossen. Door het opzetten van 'besloten gebruikersgroepen' blijft de vaak vertrouwelijke informatie beperkt tot de leden. De informatieleverancier bepaalt zelf wie toegang heeft tot zijn informatie en aan welke prijs. Het is mogelijk met de videotex-dienst toegang te krijgen of te verlenen tot uw eigen of andere computers. Daarenboven kan op termijn een internationaal schakelnet tot stand komen.

"Videotex is een ideaal publicitair medium voor grote verspreiding met attractieve grafische mogelijkheden. De informatieleverancier kan op eenvoudige en efficiënte manier zijn informatie te koop aanbieden."

Tot zover de RTT.

Nu de praktijk. Want het lijkt erop als je het RTT-proza leest dat er hier een nieuw, veelzijdig en uitgebreid medium, ter van iedereen gepaard te

beschikking staat. Zonder gaan met problemen. Niets is minder waar. Er waren problemen in de vorm van het zich door de RTT toegeëigende monopolie op de modem, het apparaat dat gegevens van digitale in analoge vorm omzet, zodat het via de telefoonlijnen naar elders gestuurd kan worden, waar ook een modem staat, die het proces in omgekeerde volgorde doet. Het monopolie werd al in het 'Journal des Tribunaux' aangevallen (12-2-1983). De wetteksten hebben het enkel over een monopolie van de RTT op de telefoonlijnen, en niet op de apparaten die op die lijnen aangesloten kunnen worden.

Helemaal onoverkomelijk zou dat monopolie niet zijn geweest, als er maar goede modems geleverd werden, en juist daar zat het probleem. Terwijl een bedrijf als Barco schermen en terminals levert met een ingebouwde modem (Barco leverde een duizendtal viewdata-terminals aan de London Stock Exchange, en 500 aan het privénet van Jet-Air) ter grootte van een kwart velletje A4, leverde de RTT een modem, die door een van de door ons gekontakteerde personen, (die liever niet met naam en toenaam genoemd worden omdat het vrij delicate zaken zijn, en de viewdata-wereld een zeer kleine is) als eef mastodont omschreven werd. Grote bij benadering twee velletjes A4, hoogte een half velletje A4. Daarbij komt nog dat de door de RTT geleverde modem geen specifieke viewdata modem is, maar een voor computers op lage snelheid. Op zich werkt hij wel goed, zeggen degenen die hem gebruiken, anderen weten het niet of nauwelijks. De modem staat er, en dat is alles.

Wat het nog erger maakte, was de exorbitante prijs voor de huur van de modem. Per 2 maand moesten de gebruikers 2.750 Bfr. neertellen, voor iets dat bij wijze van spreken in de kast stond. De kosten voor een in een Barco-terminal ingebouwde modem bedragen tussen de 1.000 en 2.000 frank. In het bui-

tenland zijn de kosten voor de gebruiker van viewdata aanzienlijk lager. In Frankrijk levert de PTT voor 1.500 Bfr. huur per maand, de terminal inclusief de modem. De huur van de modem in Duitsland bedraagt 5 mark per maand, in franken 100. De vraag was ook wat men op het kabinet van de eerste minister zou doen, nu dat uitgerust wordt met 100 videotex-aansluitingen. Ingebouwde of externe, lees RTT-modems? Aanvankelijk wisten de andere gebruikers niet beter dan dat er RTT-modems op de Barco-terminals geschakeld zouden worden. Helaas, zei men toen, omdat zij anders als er moeilijkheden zouden komen naar de handelwijze van de eerste minister

konden verwijzen. Onbekend is of het rekensommetje voor de huur alleen van de 100 modems op het kabinet van Martens de RTT ertoe aangezet heeft het apparaat vrij te geven. De dienst van de eerste minister zou dan 3.300.000 bfr. modemhuur moeten opbrengen.

Het gebruik van de RTT-modem was ook redelijk omslachtig. Men neme de telefoonhoorn in de handen, drukke op een knop, wachte tot men het signaal van de komputere hore. Dan de knop van de terminal indrukken, en de hoorn wegleggen. Dan moet nog het password ingegeven worden. Makkelijk is anders. De apparaten met ingebouwde modem hebben autodialing en geven automatisch het password. Natuurlijk wist de RTT dat het met een ingebouwde modem makkelijker ging. Op de mediadag van de CVP, 4 december vorig jaar, had Barco volledig volgens de regels op de stand externe modems. Iets verderop stond de RTT met een Barco-terminal, maar zonder externe modem! Toen een van de andere gebruikers van videotex op een show stond (opnieuw geen namen) was de modem besteld. Men had daar twee mogelijkheden, of de externe RTT-modem, of de ingebouwde. De ingebouwde was wel veel sneller aangesloten, maar een technicus van de RTT kwam daar om de externe modem aan te sluiten. De technicus belde naar de RTT om te horen wat hij moest doen. Onze contactpersoon hoorde hem zeggen: *"Als hij betaald is, is er toch geen enkel probleem?"* Voor de duur van de tentoonstelling heeft de RTT-modem ongebruikt in de kast gestaan. Maar hij was betaald.

De verplichting de RTT-modem te gebruiken was ook een rem op een plan van Test-aankoop. Test-aankoop wilde op enkele plaatsen, bijvoorbeeld City 2, een corner-operator zetten, aangesloten op de data-

Videotex in Belgium

bank van Test-aankoop met konsumenteninformatie. Naar analogie ook van de viewdata automaten die in een groot aantal Nederlandse postkantoren te vinden zijn. Test-aankoop kon het echter niet doen, want de modem trok een streep door de rekening. Overigens, viewdata-automaten zijn ook niet te vinden in Belgische postkantoren.

Blijft het in vergelijking met de modem relatief kleine probleem van de telefoonkosten. In de zones 02 en 03 (Brussel en Antwerpen) geldt het lokale tarief, omdat daar de databanken geplaatst zijn (onder andere die van Editel in Brussel, en die van Bell-Telephone in Antwerpen).

daarbuiten geldt het tarief van 5 fr. per 48 seconden. Het komt erop neer dat de gebruikers meer aan de RTT in de vorm van telefoonkosten betalen dan aan de bedrijven waarvan zij informatie vragen (Editel bijvoorbeeld vraagt 3 fr. per minuut komputergebruik). In Engeland heeft 60 tot 70 % van de viewdata-gebruikers lokaal tarief.

Het hoeft dan ook geen betoog dat de markt van viewdata tot nu toe in België erg klein is. Partikulieren hebben er voor gepast 2.750 frank alleen voor de huur van de modem te betalen, daarbovenop komen dan nog de kosten voor de telefoon en het gebruik van de komputer. Viewdata wordt nu praktisch alleen in het bedrijfsleven benut, maar ook daar zijn de

gebruikers op de vingers van twee handen te tellen: Editel (VUM), Bell Telephone, 3 Suisses, Test-aankoop, Jet-Air (echter buiten het videotex-systeem van de RTT), Telemedia, GBM, het Gemeentekrediet, Mediatel, en Sony (experimenteel). Geschat wordt dat er zo'n 200 tot 250 terminals geïnstalleerd zijn (Jet-Air niet meegeteld). Het gebruik van viewdata is voor de Regie TT nauwelijks kostenverzwarend. De telefoonlijnen zijn er, de ingebouwde modems zijn er ook.

Feitelijk hoeft de RTT alleen maar te inkasseren. Maar het vermoeden bestaat bij een enkeling dat dat nog te maken heeft met de nasleep van de Baudrin-affaire. Sinds die affaire zou niemand bij de RTT zijn nek nog uit durven steken. Pas als er iets zwart op wit van hogerhand komt, gaat het apparaat draaien.

Aan de viewdata-dienst bij de RTT ligt het niet. Een door ons geïnterviewde zei dat de dienst de dingen goed ziet zitten, en als ze eenmaal het groene licht zullen krijgen, dat alles dan goed in elkaar zit, omdat ze geleerd hebben van de ervaringen in Engeland, Nederland en Duitsland. Dat is dan een magere troost.

AVP

bron: Pub 1/6/83

16 couleurs char

```

1      REM
3      REM
10     REM *****
20     REM *** S/P 16 COULEURS CARACTERES. LIGNES      ****
30     REM *** ADRESSEES PAR LA FN CURSOR(X,Y)         ****
40     REM *** SANS CALCULS COMPLIQUES !               ****
50     REM ***      (c) ALAIN MARIATTE 1983             ****
60     REM *****
70     REM
80     REM
85     MODE 0:PRINT CHR$(12):CF!=0.0:COLORT CF! 10 CF! CF!
87     PRINT :PRINT
100    INPUT "LIGNE TEXTE A ECRIRE:";A$:PRINT
110    INPUT "CURSOR (X,Y):";X!,Y!:PRINT :CURSOR X!,Y!:GOSUB 1000
111    REM
115    REM MODE 16 COULEURS (de CA a FA selon la taille)
120    POKE CB,#FA
121    REM
122    PRINT A$
125    REM
127    REM ON CHOISIT DES COULEURS ALEATOIRES
130    FOR I!=1.0 TO LEN(A$)
140    POKE AC-3,(RND(16.0) SHL 4.0)+CF!
150    AC=AC-2
160    NEXT
170    REM
900    GOTO 100
990    REM
992    REM DETERMINER L'ADRESSE DU CONTROL BYTE & DU CURSEUR
994    REM
1000   CB=PEEK(#79)*256+PEEK(#78)
1020   AC=PEEK(#73)*256+PEEK(#72)
1030   RETURN

```

cassette tape lister

Als U, net als ik met een 'poor man's outfit' - bestaande uit: DAI pc, TV, Audio Cassette recorder (en ASR Teletype) - moet werken, dan is bovenstaand programma mogelijk ook voor U interessant. Zeker, als U evenals ik de neiging hebt om vrij veel programma's op EEN bandje te zetten, waarna het punktuueel noteren van WAT WAAR staat nogal eens te wensen over laat in de hitte van de strijd...

Genoemd programma is gebaseerd op een eerder in DAInamic verschenen List Programma, maar nu uitgebreid met een AUTOMATISCHE TELLERSTAND Berekening.

Vooraf dit laatste is m.i. de grote kracht van deze versie, omdat alleen daarmee een werkelijk bruikbare Catalogus van een Bandje wordt verkregen.

HOE WERKT 'T?

Om de inleestijd niet al te groot te laten worden, heb ik uitvoerige tekst en uitleg in het programma zelf achterwege gelaten.

Daarom hier een toelichting op het principe:

Laat de Computer de TITELS van de programma's e.d. lezen, bereken de tijd tussen het lezen van twee opeenvolgende Titels, en leidt hieruit de bereikte Tellerstand af.

Omdat de Timer in de DAI niet loopt tijdens het eigenlijke inlezen, moeten we hiervoor corrigeren door ca 2.5 s bij te tellen voor elke ingelezen titel (ca 2.3 s synchronisatietoon + de tijd nodig voor 't inlezen van de Titel. Zie regel 310, waar de verstreken tijd in 20ms eenheden wordt berekend).

Daar de Afwikkelspoel geleidelijk aan sneller gaat draaien, moet de gemeten tijd omgerekend worden met de (benaderings-) formule voor de Tellerstand 'N' zoals deze voorkomt in regel 320, resp. 490.

Hierin is 'R' de straal van de VOLLE Afwikkelspoel; 'D' de Band-dikte voor een C60 cassette; (voor een C90 is dit ca 13 μ m); en 'L' de TOTALE Lengte van de tape, gerekend vanaf het Begin van de Band (Tellerstand 000!) tot aan het begin van het Programma waarvan we de Tellerstand 'N' willen berekenen (zie ook regels 200 t/m 220).

De waarde van 'L' volgt uit: $L = T * V! * 2E-2 * F!$

Daarbij is 'T' de in regel 310 berekende verstreken tijd in eenheden van 20 ms; 'V' is de bandsnelheid (in m/s), en 'F' een correctie factor, welke afhangt van het betreffende cassettebandje - zie hierna (aanvankelijk is de waarde van 'F' 1.0, reden waarom hij werd weggelaten in regel 320).

De faktor '0.6' is de overbrengingsverhouding tussen de Afwikkelspoel en de Teller (mogelijk is deze waarde afwijkend voor een ander merk dan de Philips N2225!)

Als we het programma starten zal dit eerst naar de Datum vragen, vervolgens naar de Naam van het Bandje, en tenslotte of het al een 'EOF' bevat. Als we n.l. achter het laatste programma op de band een z.g. "End-Of-File Record" hebben staan dan kan daar op getest worden om te stoppen. (Het is anders niet (eenvoudig) mogelijk erachter te komen dat we het laatste programma op de band 'gezien' hebben..)

Ontbreekt het EOF record, dan biedt de computer aan er een voor ons te schrijven. Wensen we dit niet, of zijn we slechts geïnteresseerd in een beperkt aantal titels, dan kunnen we in plaats daarvan het Aantal Programma's opgeven.

Nadat we het bandje geheel teruggespoeld hebben en de TELLER op 000 hebben gezet, moeten we de Recorder starttoets en de Spatiebalk gelijktijdig indrukken. Het programma leest nu alle Titels, toont ze op het scherm, en

registreert zowel de verlopen Tijd (T), de berekende Tellerstand (N) als de gelezen Titel (A\$) in de daartoe bestemde Arrays ("T(I)", "N(I)", "T\$(I)").

Als het gevraagde aantal Titels, resp. 'EOF' is gelezen, wordt een overzicht van alle geregistreerde titels getoond met hun Berekende Tellerstanden voor F!=1.0.

Vervolgens wordt gevraagd wat de WERKELIJKE stand behorend bij het als laatste getoonde Programma (resp. 'EOF') is. Hiermee wordt dan een iteratieve herberekening gemaakt (aanpassing van 'F!') om de zaak kloppend te maken (soms lukt dit niet precies voor alle tellerstanden, maar meer dan 1 à 2 nummers scheelt het zelden!).

Na de berekening wordt ter controle het gekorrigeerde overzicht weer getoond, en kan men desgewenst opnieuw een 'werkelijke waarde' ingeven, enz.

Tot slot moet 'O' worden gegeven om uit deze Loop te raken en komt de vraag: "Print-out op de ASR (Y/N)?"

I.g.v. "Y" volgt de waarschuwing om de ASR On-Line te zetten, en wordt er na het indrukken van de Spatiebalk een Print-out gemaakt (zie voorbeeld).

N.B. Mocht men moeite hebben met de Lineariteit (nummers in middengebied te hoog of te laag), dan moet mogelijk de waarde van 'R!' (en/of 'D!') iets aangepast worden! (regel 240). Ook kan het zijn dat de bandsnelheid 'V!' niet precies 4.75 cm/s is.

Dit 'tunen' kan men heel eenvoudig doen door tijdens een 'BREAK' een of meer waarden te wijzigen, en na 'CONT' opnieuw de juiste 'werkelijke' waarde in te geven! (Zonodig herhalen tot de gewenste nauwkeurigheid is bereikt..)

Overigens kan men het uitprinten natuurlijk op elke willekeurige Printer laten doen. In dat geval moet alleen de Baudrate worden aangepast, en is het wachten na een Carriage Return (Lege Loop met 'NUMBER' - geen WAITTIME, daar anders de timing verloren gaat!) meestal niet meer nodig.

cassette tape lister

83-05-20 PAG 1

```

10  REM CASSETTE-TAPE LISTER  V2.4
20  REM DØØP D. ASSINK; 83-05-20
30  CLEAR 15000: DIM X(0), N(100), T(100), TS(100)
35  EL$=""
40  CURS=20: AANT=100: I=1: BAUD=#CO: PØKE #FF05, BAUD
50  PØKE #131, 0: PRINT CHR$(12); : ADS="" : X9$=CHR$(9)
60  RESTØRE: FØR A=1 TØ 21: READ B: ADS=ADS+CHR$(B): NEXT
70  AD=PEEK(VARPTR(ADS)) IØP (PEEK(VARPTR(ADS)+1) SHL 8)+1
80  CURSØR 15, 20: PPINT "CASSETTE TAPE LIST PRØGPAMMA"
90  PRINT TAB(15); "===== ": PRINT : PRINT
100 PRINT "DIT PRØGRAMMA STELT DE INHØUDSØPGAVE VAN
    EEN CASSETTEBANDJE"
110 PRINT "SAMEN; WAARNA DIT ØP EEN ASF TELETYPE GEPRINT
    KAN WØRDEN. ": PRINT
120 PRINT "EEN EVT. AFWIJING TUSSEN DE BEREKENDE EN DE WERKELIJKE"
130 PRINT "TELLERSTAND KAN NØG VØØR HET UITPRINTEN WØRDEN
    GECØRRIGEERD."
140 PRINT : PRINT : IF T=0 THEN INPUT "DATUM"; D$: PRINT
150 PRINT : INPUT "NAAM ØF NUMMER VAN HET BANDJE"; N$: PRINT
160 PPINT : INPUT "IS HET BANDJE AL VØØRZIEN VAN 'EØF' RECØRD (Y/N)";
    A$: PRINT
170 IF A$<>"Y" THEN GØSUB 700
180 T=0: PRINT CHR$(12): PØKE #75, #20
190 PEM
200 PEM NU VØLGT DEFINITIE VAN: 'R' (RADIUS VØLLE SPØEL [MM]);
210 PEM 'D' (DIKTE BAND [MICRØN] [C60]); 'V' (BANDSNELHEID [CM/S])
220 PEM EN: 'F' (EXPERIM. CØPRECTIEFAKTØR VØØP GEM. BANDJE):
230 R!=24.7: D!=16.2: V!=4.75: F!=1.0: FF!=F!
240 R!=R!*1E-3: D!=D!*1E-6: V!=V!*F!*1E-2
250 CURSØR 0, 23: PRINT "START RECØPDER, EN DRUK TEGELIJK
    ØP SPATIEBALK!"; : GØSUB 600
260 B=#BFE7+239: PØKE #1BE, 250: FØKE #1BF, 255: CURSØR 0, 21
270 PRINT " NR: TYPE: FILE NAAM:"
280 CURSØR 0, 23: PRINT EL$;
290 CURSØR 0, 23: CALLM AD
300 TL=PEEK(#1BE): TH=PEEK(#1BF): PØKE #1BE, 255: PØKE #1BF, 255
310 T=T+255-TL+113+0.42*(#BFE7-B)+(255-TH)*256: REM TIMING
320 T(I)=T: L!=T*V!*2E-2: N=(R!-SQR(F!*R!-D!*L!/PI))/D!*0.6
330 N(I)=N: CURS=CURS-1: IF CURS=0 THEN CURS=19
340 CURSØR 0, CURS: PØKE #40, #30
350 PRINT EL$; : CURSØR 0, CURY: PRINT N; TAB(6);
360 A=#BFE7: A$=CHR$(PEEK(A))+ " "
370 A=A-2: IF PEEK(A)=#20 GØTØ 370
380 FØR B=#BF85 TØ A STEP 2: IF PEEK(B)=#20 THEN NEXT
390 IF A>B THEN FØR A=A TØ B STEP -2: A$=A$+CHR$(PEEK(A)): NEXT
400 TS(I)=A$: PRINT A$: I=I+1: IF I<=AANT AND FIGHT$(A$, 3)<>"EØF"
    GØTØ 280: GØTØ 410
410 PØKE #FF05, BAUD: IF BAUD=1 THEN WAIT TIME 50
420 PRINT CHR$(12): PRINT "BAND: "; N$; TAB(45); "DATUM: "; D$: PRINT
430 PRINT " NR: TYPE: FILE NAAM: ": PRINT : IF BAUD=1
    THEN WAIT TIME 20
440 FLG=0: FØR K=1 TØ I-1: T=T(K): L!=T*V!*2E-2*F!
450 N=(R!-SQR(R!*R!-D!*L!/PI))/D!*0.6
460 PRINT N; TAB(6); TS(K): FØP J=1 TØ NUMBER: NEXT: NEXT K
470 PRINT : IF BAUD=1 GØTØ 560: INPUT "WERKELIJKE WAARDE VAN
    LAATSTE NUMMER (0 = EXIT)"; N: PRINT
480 IF N=0 GØTØ 530: M=N: K=M: T=T(I-1)
490 F!=FF!*K/N(I-1): L!=T*V!*2E-2*F!: N=(R!-SQR(R!*R!-D!*L!/PI))/D!*0.6
500 IF N<M THEN K=K+1: FLG=1: GØTØ 490
510 IF N>M THEN IF FLG=0 THEN K=K-1: GØTØ 490
520 GØTØ 410
530 PPINT : INPUT "PRINT-ØUT ØP DE ASF (Y/N)"; A$: PPINT

```

```

540 IF AS="Y" THEN GOSUB 800:GOTO 410
550 GOTO 570
560 POKE #131,1:PRINT "ZET ASF OFF-LINE!!!":GOSUB 810
570 NUMBER=0:BAUD=#CO:POKE #131,0:GOTO 40
600 IF GETC<>0 GOTO 600
610 IF GETC=0 GOTO 610:RETURN
700 PRINT :INPUT "'EOF' RECØRD SCHRIJVEN";AS:PPINT
710 IF AS<>"Y" GOTO 770
720 PRINT :PRINT "SPØEL CASSETTE NAAR GEWENSTE PØSITIE"
730 PPINT :PRINT "SET RECØRD,START TAPE,TYPE SPACE"
740 GOSUB 600:SAVEA X "EOF":PRINT
750 PRINT "SPØEL CASSETTE TERUG NAAR 000, EN TYP EEN SPATIE"
760 GOSUB 600:GOTO 780
770 PRINT :INPUT "HØVEVEL PRØGRAMMA'S STAAN ER DAN (TENMINSTE) ØF";
AANT:PRINT
780 RETURN
800 BAUD=1:NUMBER=1500:PRINT :PPINT "ZET ASF ØN-LINE!!!"
810 ENVELOPE 0 15,15;0,12;:SØUND 0 0 15 0 FREQ(2000)
820 GOSUB 600:SØUND ØFF :RETURN
10000 DATA #F5,#C5,#D5,#E5,#01,#40,#00,#11,#B1,#80,#21,#9E
10010 DATA #E6,#CD,#CE,#02,#E1,#D1,#C1,#F1,#C9
    
```

Onderstaand een Voorbeeld van de Output zoals die op het scherm, en uiteindelijk op de Printer verschijnt:
 (Bandje bevat een 'EOF' record op tellerstand 100)

BAND: DEMØNSTRATIE CASS.TAPE LISTER

DATUM: 83-05-20

NR: TYPE: FILE NAAM:

```

5 0 CASSETTE TAPE LISTER V2.4
9 0 TEST PRØGPAMMA 1
14 0 TEST PRØGPAMMA 2
19 2 ARRAY XYZ
24 2 ARPAY PQR
29 0 BUDGET
38 0 PLANNING V1.1
48 2
57 0 MAIL V2.3
86 0 DØØLHØF V2.2
96 2 EOF
    
```

← De Output zoals die de eerste maal op het SCHERM verschijnt.

WERKELIJKE WAARDE VAN LAATSTE NUMMER (0 = EXIT)?100

BAND: DEMØNSTRATIE CASS.TAPE LISTER

DATUM: 83-05-20

NR: TYPE: FILE NAAM:

```

5 0 CASSETTE TAPE LISTER V2.4
10 0 TEST PRØGPAMMA 1
15 0 TEST PRØGPAMMA 2
20 2 ARRAY XYZ
25 2 APRAY PQR
30 0 BUDGET
40 0 PLANNING V1.1
50 2
60 0 MAIL V2.3
90 0 DØØLHØF V2.2
100 2 EOF
    
```

← De HERBEREKENDE Output zoals die na ingave van de juiste tellerstand (hier dus: 100) voor het 'EOF' record ontstaat. Door hierna als 'werkelijke' waarde '0' te geven, wordt deze zelfde tekst uitgeprint.

>>>>> Programmeer technieken <<<<<<<

Het probleem dat ik deze keer wilde bespreken is het toekennen van een waarde aan een variabele als de toegekende waarde weer op zijn beurt afhankelijk is van de waarde van een andere variabele. Dit komt in vele programma's voor en wordt vaak opgelost op de manier die ik hieronder laat zien :

```
150   IF A=5 THEN P=3:GOTO 200
160   IF A=6 THEN P=5:GOTO 200
170   IF A=7 THEN P=7:GOTO 200
180   IF A=8 THEN P=9:GOTO 200
190   P=0
200   .....
```

Dit is vanzelfsprekend een duidelijke methode, die zelfs in sommige gevallen de beste is. Maar als er veel mogelijkheden voor A en P zijn of als de toewijzing met een berekening uit te voeren is verdient dat toch de voorkeur, mits die berekening niet te ingewikkeld is.

Ik zal een aantal uitgewerkte voorbeelden geven hoe het in de meest voorkomende gevallen beter opgelost kan worden. Het betere van die oplossing zal een korter programma dat vaak sneller zal zijn; al hoeft dit laatste niet. In tegendeel het programma zal zelfs trager kunnen worden; als normaal in bijna alle gevallen reeds aan de eerste of tweede conditie wordt voldaan zal mijn aanpak trager kunnen zijn.

Kiest u voor de 'oude' oplossing doe dit dan wel op een overdachte manier. Het volgende voorbeeld zal dit duidelijk maken. We willen de getallen van 100 tot en met 300 onderzoeken op deelbaarheid door de priemgetallen onder de 20. (Hiermee vinden we de priemgetallen tussen deze grenzen. (19 is overbodig)) We gebruiken hiervoor het volgende programma: (intikken na IMPFPT)

```
5     WAIT TIME 1:POKE #1BE,#FF:POKE #1BF,#FF
10    FOR I=100.0 TO 300.0
20    IF I/2.0=INT(I/2.0) GOTO 110
30    IF I/3.0=INT(I/3.0) GOTO 110
40    IF I/5.0=INT(I/5.0) GOTO 110
50    IF I/7.0=INT(I/7.0) GOTO 110
60    IF I/11.0=INT(I/11.0) GOTO 110
70    IF I/13.0=INT(I/13.0) GOTO 110
80    IF I/17.0=INT(I/17.0) GOTO 110
90    IF I/19.0=INT(I/19.0) GOTO 110
100   PRINT I
110   NEXT
195   A=PEEK(#1BE):B=PEEK(#1BF):?("#FFFF-A-B*256.0)/50.0;" SEC"
```

Zoals u hopelijk weet zijn regel 5 en 195 bedoeld om de looptijd van dit programma te meten. Op mijn machine deed dit programma er 12.44 (6.46) seconden over. De tijd tussen haakjes is de tijd met math.chip aan. We zien simpel in dat bij de helft van de getallen reeds bij regel 20 naar de next gesprongen wordt. De hier gekozen testvolgorde is dus juist en logisch. Als we nu voor de demonstratie de regels 20 t/m 90 eens in omgekeerde volgorde nummeren zullen we zien dat de tijd enorm toeneemt nl 21.74 (11.12) seconden.

Men kan dit programma natuurlijk ook wel beter opzetten: (intikken na IMPINT)

```
5   WAIT TIME 1:POKE #1BE,#FF:POKE #1BF,#FF
10  FOR I=100 TO 300
20  IF I/19*19=I GOTO 110
30  IF I/17*17=I GOTO 110
40  IF I/13*13=I GOTO 110
50  IF I/11*11=I GOTO 110
60  IF I/7*7=I GOTO 110
70  IF I/5*5=I GOTO 110
80  IF I/3*3=I GOTO 110
90  IF I/2*2=I GOTO 110
100 PRINT I
110 NEXT
195 A=PEEK(#1BE):B=PEEK(#1BF):PRINT (#####-A-B*256)/50.0
```

Werkt dit dan ook zullen sommigen zich misschien verwonderd afvragen? Inderdaad dit werkt ook: als we een getal I eerst integer delen door bv 17 en dan weer vermenigvuldigen met 17 zal het getal afgerond worden op het grootste veelvoud van 17 dat kleiner of gelijk is aan het getal I. Als I dus deelbaar is door 17 zal aan de conditie $I/17*17=I$ voldaan worden. Dit programma duurt 9.58 (6.56) seconden. De winst komt mijns inziens voornamelijk doordat er integer wordt gerekend, maar dit is niet te controleren omdat de 'truc' in floating point niet werkt. We zien echter wel een vreemde volgorde; we zetten dus de regels 20 t/m 90 weer in omgekeerde volgorde en krijgen 5.88 (4.4) seconden. Gaan we nu regel 30 bij regel 20 zetten en evenzo 50 bij 40 etc. komen we op 5.82 (3.74) seconden. Maar het kan nog steeds beter:

```
20  IF I/2*2=I THEN NEXT:GOTO 195
30  IF I/3*3=I THEN NEXT
40  IF I/5*5=I THEN NEXT
50  IF I/7*7=I THEN NEXT
60  IF I/11*11=I THEN NEXT
70  IF I/13*13=I THEN NEXT
80  IF I/17*17=I THEN NEXT
90  IF I/19*19=I THEN NEXT
100 PRINT I:NEXT
```

Regel 5,10 en 195 identiek aan vorige keer. Minder fraai doordat er bij een FOR maar liefst 9 NEXT'n staan en tevens reeds van te voren bepaald is dat het laatste getal door 2 deelbaar is. Om het programma algemeen te houden zou achter elke NEXT een GOTO 195 moeten staan. Maar dit zou alleen de intiktijd doen toenemen en niet de looptijd 5.7 (3.62) seconden. Er is echter nog winst te behalen door regel 100 te veranderen in:

```
100 PRINT I;:NEXT
```

Een kleine verandering maar de puntkomma achter de PRINT scheelt veel, we krijgen nu 4.96 (2.88) seconden.

U ziet een math chip versnelt wel (30 tot 50 %) maar goed programmeren loont vaak meer (75 %).

We zijn nog steeds bezig met de 'oude' versie en het wordt toch wel eens tijd om een echt andere aanpak te laten zien. Kijk nog eens naar het uitgangspunt.

Eerste probleem: A kan zijn 2, 3, 4, 5, 6, 7 of 8 en in dezelfde volgorde moet P worden 12, 15, 18, 21, 24, 27 of 30.

We zien vrij snel dat er een wiskundig verband bestaat tussen de waarde van A en de waarde van P. A loopt met 1 op als P met 3 oploopt; om dit verschil tot uitdrukking te laten komen zullen we A met 3 moeten vermenigvuldigen en om het dan geheel met elkaar in overeenstemming te brengen er nog 6 bij doen. We krijgen dan:

OUD	NIEUW
150 IF A=2 THEN P=12	
160 IF A=3 THEN P=15	
170 IF A=4 THEN P=18	150 P=A*3+6
180 IF A=5 THEN P=21	
190 IF A=6 THEN P=24	
200 IF A=7 THEN P=27	
210 IF A=8 THEN P=30	

Het oude programmadeel kan nog versneld worden door GOTO's aan het eind van de regels. Er is wel een verschil maar in de praktijk zal dat meestal niet een probleem zijn, 'oud' verandert P niet als aan geen der voorwaarden voldaan wordt en 'nieuw' verandert P altijd.

Tweede probleem: iets moeilijker nu. Haast het zelfde maar nu kan A elke waarde hebben en P moet 9 worden als A kleiner dan 2 is en 33 als A groter dan 8 is.

OUD	NIEUW
150 IF A<2 THEN P=9	
160 IF A=2 THEN P=12	
170 IF A=3 THEN P=15	150 P=A*3+6
::: :: ::: ::::: :::::	160 IF A<2 THEN P=9
220 IF A=8 THEN P=30	170 IF A>8 THEN P=33
230 IF A>8 THEN P=33	

Nadeel van de nieuwe methode is het mogelijkwerijs voor niets uitvoeren van regel 150. Als in vele gevallen A kleiner is dan 2 is het dan ook beter regel 150 en regel 160 te verwisselen waarbij er dan natuurlijk wel een GOTO 180 moet komen achter de P=9. En zo zijn er nog wel wat varianten te verzinnen in geval A meestal een bepaalde waarde veel vaker dan andere aanneemt.

[
Derde probleem: we gaan nu het geval bekijken waar A regelmatig toeneemt maar waar er geen eenvoudig wiskundig verband is tussen A en P, een berekening is dus lastig of onmogelijk.

OUD	NIEUW
150 IF A=3 THEN P=7	
160 IF A=4 THEN P=4	10 DIM P(9)
170 IF A=5 THEN P=15	20 FOR I=3 TO 9:READ P(I):NEXT
180 IF A=6 THEN P=31	
190 IF A=7 THEN P=76	150 P=P(A)
200 IF A=8 THEN P=45	
210 IF A=9 THEN P=29	900 DATA 7,4,15,31,76,45,29

De voorgestelde nieuwe methode vertraagt het programma enigzins bij aanvang maar compenseert dit later ruimschoots. Tel het aantal regels niet; het is een voorbeeld, als ik 'oud' uitbreid tot bv twintig regels zal 'nieuw' gelijk blijven of met hoogstens een toenemen.

Vierde probleem: het geval waar P regelmatig toeneemt en A zich juist zeer onregelmatig gedraagt. Ook hier zullen we in de versie 'nieuw' gebruik maken van een array.

```

      OUD
150  IF A=3 THEN P=2
160  IF A=5 THEN P=3
170  IF A=9 THEN P=4
180  IF A=12 THEN P=5
190  IF A=33 THEN P=6
      XT
200  IF A=42 THEN P=7
210  IF A=57 THEN P=8

```

```

      NIEUW
10   DIM A(8)
20   FOR I=2 TO 8:READ A(I):NEXT
150  FOR I=2 T 8:IF A(I)=A GOTO 160:NE
160  P=I

```

Vijfde probleem: er is totaal geen logica te ontdekken in de onderlinge verhoudingen van A en P nog in de waarde die zij hebben of krijgen. Dit is bijvoorbeeld

het geval als A de ASCII-code is van een bepaalde toets en P de actiecode voor het programma. We gaan in de 'nieuwe' oplossing weer met een array werken. Er kan gekozen worden voor een twee-dimensionale array waar de A's e P's in paartjes bij elkaar staan of voor twee aparte array's; een voor de A's en een voor de P's

De laatste mogelijkheid is misschien iets minder mooi maar werkt sneller.

```

      OUD
10   IF A=16 THEN P=7
20   IF A=17 THEN P=8

30   IF A=18 THEN P=15
40   IF A=19 THEN P=16
50   IF A=65 THEN P=0
60   IF A=66 THEN P=2
70   IF A=74 THEN P=-1
80   IF A=78 THEN P=99
90   IF A=83 THEN P=100
100  IF A=9 THEN P=5

```

```

      NIEUW
10   DIM A(10),P(10)
20   FOR I=1 TO 10:READ A(I),P(I):NEXT

50   FOR I=1 TO 10
60   IF A=A(I) THEN P=P(I):GOTO 80
70   NEXT
80   .....

900  DATA 16,7,17,8,18,15,19,16,65,0
910  DATA 66,2,74,-1,78,99,83,100,9,5

```

Nogmaals tel niet het aantal regels het gaat om het principe. En wat 'oud' gedaan wordt door de regels 10 t/m 100 wordt 'nieuw' gedaan door de regels 50,60 en 70. Indien gewenst kan dit ook nog in een regel gezet worden:

```
50  FOR I=1 TO 10:P=P(I):IF A=A(I) GOTO 60:NEXT
```

Toch kleven er ook wat nadelen aan de voorgestelde nieuwe methodes, waar we op moeten letten willen we er geen last van krijgen. De nieuwe regel 6 eindigt met GOTO 80; dit zal de snelheid ten goede komen maar de FOR-NEXT loop is dan vaak nog niet ten einde. Komt er dan een NEXT van een buitenloop zal die verkeerd worden geïnterpreteerd. Dit kan voorkomen worden door NEXT I ipv alleen NEXT te gebruiken. De een-regel methode zal traag zijn: alle controles met behulp van array's en indien nog niet gevonden toch maar vast een toewijzing.

Daarnaast zal juist de ongeoefende programmeur het minder duidelijk zien wat de regel doet. Voor mij is het een voordeel bij het bekijken van inzendingen: Ik kan het programma korter maken en ik weet gelijk met wat voor soort programmeur ik hier te doen heb.

Ik zie echter in de inzendingen een duidelijke stijgende lijn zitten en ik vlei me met de gedachte dat dit mede door deze artikelenreeks komt

Frank H. Druijff

P.S. Het problem program is zowel door Nico P. Looije als door mij zelf nog aanzienlijk versneld.

The EDITOR story

This article describes the functioning of the BASIC-command 'EDIT'.

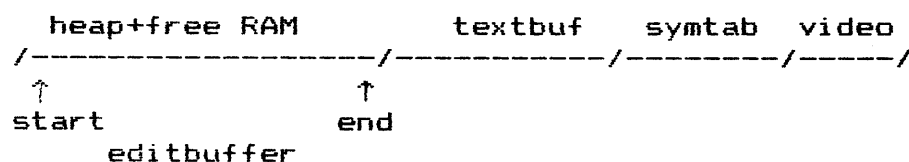
For details, the 'DAI pC FIRMWARE MANUAL' should be used.

When the Basic command 'EDIT' is given (only in direct mode), the program jumps to address #0E1F5.

- Set up the editbuffer:

E1F5 The EDIT buffer is set up. The screen mode 0 is
E265 selected (the screen is cleared). Via #D86D, an
 error message is given if insufficient memory
 for mode 0 is available.
 If sufficient memory available, the heap and the
E26C symbol table are cleared (all variables are set
 to zero). Then the free RAM space is calculated,
 and the amount of heap-space is added to it.
 This gives the maximum available free RAM space
 for the editbuffer.

E275 Now the textbuffer and the symboltable are moved
 to the end of the free RAM. This leaves a large
 free RAM area between 'start of heap' and 'start
 textbuffer':



E27D The startaddress of the heap +2 is the startaddress
 of the editbuffer; the startaddress of the text-
E28E buffer is set as the end of the editbuffer.

The output switch #0131 is set to the editbuffer.

E1F8 Now the program is listed into the editbuffer.
E19F This is done in the same way as it is listed to
 the screen. So the contents of the editbuffer
 is a complete listing of the program in 'plain
 Basic text'.

E1FC Now the output switch #0131 is set back to the
 screen. The keyboard is enabled completely, and
 a '0' byte is added at the end of the editbuffer
 contents to indicate its end (all other bytes
 are <> 0).
 The tab-table is disabled by clearing its pointer
 #00B4. Therefore, TAB's are normally not useable
 in the editbuffer.

E20B Here the screen-editor is initialised via RST5;

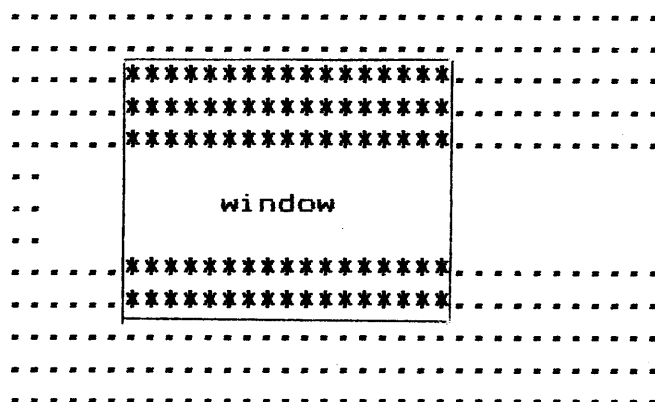
DATA #2A:

- 2EBF4 Again a mode 0 screen is set up and cleared, and all pointers for handling the cursor and the window (the visible part of the textbuffer on the screen) are cleared.
- 2EC17 On this cleared screen, one page (24 textlines) beginning at the start of the editbuffer, are printed.

- Inputs into the editbuffer:

- E20D Now the program enters a loop, in which inputs can be made. Depending on the key pressed, different actions are taken by the screen editor. Inputs can be stopped by pressing the BREAK-key.
- E216 The inputs made are evaluated via RST5; DATA #2D:
- 2EC1E Via the cursor control keys, the cursor can be moved over the screen. The screen scrolls automatically if the boundaries of the screen are reached. This means, that then the window is moved over the text in the editbuffer.

This 'window' can be seen as a rectangular looking glass, moved over a piece of paper. Only the part seen through this looking glass is visible:



The window can also be moved by depressing the cursor control keys together with the SHIFT-key. Now the cursor remains on the same text line (as long as the screen boundaries are not reached).

- 2EF4B Any other character typed in, is inserted in the editbuffer on the position of the cursor. The rest of the text is moved to make space for the new character.
- 2EFCC If the input character is 'CHAR DEL', the character at the current cursor position is removed from the buffer, and the rest of the text is moved one position.
- E218 If an input character has been evaluated and eventually stored in the editbuffer, the next input character is awaited and obeyed. This process

continues until the 'BREAK' key is pressed.

- Return from EDIT:

- E21B After a 'break', the screen is cleared. Then the next key input is awaited.
- E222 If the second key depressed is again 'BREAK',
E24D all corrections made on the text in the editbuffer is ignored. In other words: the original program is restored.
The pointers for the heap, the textbuffer and the symbol table are set back to their original values, and the program is moved to its original position.
The editbuffer pointers and the contents of the editbuffer are not cleared. But because the editbuffer was loaded from the lower end of the free RAM (incl. heap) onwards, it is overwritten by the original program.
- E225 If after the first 'break', the second key is any other key except the BREAK-key, the modifications implemented in the editbuffer are now inserted in the original program.

- Insert edited program lines:

- E22A The encoded input switch #0135 is set to the editbuffer. This means: the editbuffer is the source to get inputs for encoding a textline.
- E231- Because in the pointers #0119 and #011B the start
E238 and the end of the part of the program which was listed into the editbuffer was kept, it is now possible to removed this part from the original program (which is still at the upper end of the free RAM space).
- E23B- Now the length of the area occupied by the used
E247 part of the editbuffer is calculated: from the startaddress of the heap (= start editbuffer) to the inputpointer #00A4 of the editbuffer).
- E248 Now the original program (minus the lines deleted) is moved to the end of the editbuffer:

```

      editbuf      textbuf      symtab      free RAM
/-----/-----/-----/-----/
↑
start heap

```

- E24B Now the Basic subroutine 'EDIT' is finished, and the program returns to the normal Basic monitor routine on #C818.
- C84B Because the encoded input switch #0135 is set to the editbuffer, textlines are readed from this
D879 buffer.

Geachte Heer,

Ondergetekende zendt U bij deze een complete sourcelisting van het beloofde programma, dat het mogelijk maakt dat alle microcomputers, welke werken op basis van de 8080,8085 of Z80 microprocessor kunnen communiceren met de DAipc in machinetaal via de audio cassette-recorder.

Het complete programma bevat de schrijf- en leesroutines, die via de originele DAipc-interface een machinetaalprogramma op cassette kunnen opnemen (CASSrc) en van cassette kunnen lezen (CASSrd), indien het micro-systeem van dezelfde interface wordt voorzien.

Wanneer de adressen waar het programma zich bevindt in een ander systeem voor andere doeleinden gebruikt worden, is het uiteraard mogelijk het geheel te verplaatsen. Dit zal voor de meeste DAipc-gebruiker die de DNA-assembler kunnen hanteren niet moeilijk zijn.

Het nut van dit programma komt goed tot uiting in onderwijs situaties, waar de DAipc draait als hulpmiddel voor het ontwikkelen van machinetaalprogramma's door middel van de DNA-assembler (of de SPL-macro-ass.) De door de DNA-assembler gegenereerde objectfiles m.b.v. #P. kunnen opgenomen worden op cassette en daarna in het micro-systeem via de leesroutine worden ingelezen. Dit bespaart de gebruiker het saai intoetsen van de hex-codes.

Ondergetekende denkt hiermee vooral de DAipc-gebruikers met een technische inslag een plezier te kunnen doen.

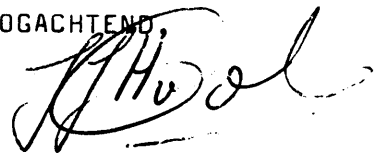
Mocht er in de toekomst belangstelling zijn voor een CHECK programma voor het testen van een opgenomen programma, dan zou ik dit gaarne van U horen.

Reeds eerder hebt U het schrijfprogramma toegezonden gekregen, daar het leesprogramma nog niet beschikbaar was. Dat schrijfprogramma kan worden vernietigd, daar het nu geleverde programma iets gewijzigd is.

Met genoegen ziet ondergetekende plaatsing in DAInamic tegemoet.

Met vriendelijke groeten van een noorderbuur,

HOOGACHTEND,



J.J.H. van OoL
Schierstins 19
7608 XZ Almelo. NL
Leraar electronica aan
de CHR. M.T.S te Almelo.

8080 cassette routines SDK-85

CASSrc V1.0 and CASSrd V1.0 on the SDK 85 microcomputerkit.

Before you can make use of the routines CASSrc and CASSrd in the SDK 85 microcomputer, it is necessary to program the routines in an EPROM (8755A) and install it at position A15 of the SDK 85 kit.

Further on the microcomputerkit has to be provided with the same cassette-interface as the DAIPC wich has to be connected to port 23H at b0 and b1.

The CASSrc routine makes it possible to record a Machine Language Program (MLP) on an audio cassette recorder in the same way DAIPC does. A MLP wich has been recorded from the DAIPC on audio-cassette, can be read into the SDK 85 microcomputerkit with the CASSrd-routine.

The way of recording and reading programs you can study in DAInamics Newsletter no. 14 and 15. (Part 1 and 2 from Mr. Jan Boerrigter)

The schematics of the cassette-interface you can find in DAInamic no.13 page 372.

The CASSrc and CASSrd routines are very usefull for anyone who wants to create objectprograms for microcomputersystems and therefore makes use of the DAInamic Assembler or perhaps Sphinx Macro Assembler on the DAIPC (eg. in technical tuition)

The routines can be easily made suitable for other microcomputersystems, based on 8085 compatible processors like 8080 and Z80.

How to use the CASSrc-routine ?

First you have to insert some data in the SDK 85 with the SUBST.MEM.-function on addresses you can find in the heading of the sourcelisting.

The following data have to be inserted;

NAME : max. 16 hex.dec ASCII characters

NAMElength (NMLG)

Lowest address of the MLP wich has to be recorded (LADR)

Destination address of the MLP (DADR). The destination address can be the same as the LADR, but it is also possible to load a program on other locations in other microsystems. So the DADR has to be different from the LADR.

Highest address of the MLP (HADR).

Program starting;

- audio-recorder on 'record' and start recorder.
- Type in : (GO) 0800 (EXEC)

Then the name of the routine (CASSrc) is shown in the SDK 85 display during recording of the MLP.

When recording is ready the display shows -80 85.

When the NAMElength is longer than 16 there is an error-reporting (-Err rc) in the display and no recording takes place.

At this moment there isn't a CHECK-routine available, but of course it is possible to check your recording on the DAIPC with the CHECK-command in BASIC.

How to use the CASSrd-routine ?

Before starting the CASSrd-routine it is always necessary to insert the OFFSET with the SUBST.MEM.-function. When OFFS = 0000H, then the MLP is loaded on the address wich is equal to DADR on tape. When OFFS has a higher value the loading address becomes DADR + OFFS. So it can be adjusted to the microsystem wich is in use.

Program starting;

- Type in : (GO) 0900 (EXEC)
- Start recorder at the beginning of the MLP that you want to load.

After starting the CASSrd-routine it displays its own functionname during reading (loading) the MLP and when there are no reading-errors (-Err rd) the reading is ready if -80 85 appears in the display of the SDK 85. When there is an error-reporting in the display (-Err rd) the loading is interrupted from the moment the fault has appeared. Check the volume-level on your recorder !

When there is a -Err rd error-reporting during reading your MLP, you must be sure, that the name of the program wasn't too long at the moment you recorded your MLP on DAipc. Otherwise it is not possible to load the program.

After loading the MLP without errors it is possible to examine all the data with the SUBSI.MEM.-function like NAME, NMLG, LADR, DADR, HADR, OFFS on the addresses you can find in the heading of the CASSrd-sourcelisting.

J.J.H. van OoL
Schierstins 19
7608 XZ Almelo.
Nederland

bootstrap for screen files

```
200  MODE 5A:PRINT CHR$(12):COLORT 8 0 0 8:COLORG 0 5 10 15
210  POKE #75,#5F:POKE #74,2:CLEAR #100
220  PRINT "*** DAINAMIC BOOTSTRAP LOADER V1.0 ex.***":PRINT
230  READ BGNADDR%,NMBR%
240  FOR ADDR%=BGNADDR% TO BGNADDR%+NMBR%-1
250  READ BYTE%:POKE ADDR%,BYTE%
260  NEXT:READ C%:IF C%<>#FFFF THEN PRINT "DATA READ ERROR":PRINT :END
270  PRINT "[ LOADING UTILITY FILE 1]"
280  CALLM BGNADDR%
310  DATA #F800,95
320  DATA #31,#00,#F9,#21, #00,#00,#22,#00, #01,#01,#FF,#31, #CD,#CE,#02,#21
330  DATA #5D,#F8,#11,#5F, #F8,#CD,#D1,#02, #21,#00,#C0,#EB, #2A,#5D,#F8,#DC
340  DATA #D1,#02,#CD,#D4, #02,#D2,#A8,#D2, #21,#00,#01,#22, #9D,#02,#21,#EC
350  DATA #02,#22,#9B,#02, #CD,#B8,#DE,#CD, #5E,#DD,#01,#40, #F8,#C3,#92,#C8
360  DATA #AD,#01,#20,#18, #11,#5B,#20,#4C, #4F,#41,#44,#49, #4E,#47,#20,#42
370  DATA #41,#53,#49,#43, #20,#5D,#FF,#8B, #19,#00,#AD,#00, #87,#00,#00,#FFFF
```

```
10  COLORG PEEK(#BFFE) IAND #F PEEK(#BFFA) IAND #F PEEK(#BFF6) IAND #F
PEEK(#BFF2) IAND #F
20  IF PEEK(#BFF1)=#FF THEN MODE 5:GOTO 30
25  POKE #9D,#B:MODE 6
30  GOTO 30
```

```

002 *****
003 *BRON      :8080 DNA-ASSEMBLER CHR.MTS. ALMELO.
004 *BESTEMMING :SDK 85  6.144 MHz CLOCK (T=326 nSEC)
005 *NAAM      :van OOL
006 *DATUM     :07-06-1983
007 *BUFFERDIMENSIE ASSEMBLER : 12
008 *TELLERSTAND BANDOPNAME VAN 004 TOT 020
009 *****
010 *TOELICHTING:
011 * HET PROGRAMMA SCHRIJFT DATA NAAR AUDIO-TAPE
012 * VOLGENS DE DAIPc-METHODE, ALS VOLGT:
013 *
014 *--VOORSIGNAAL (LEADER) CA. 1780 Hz (10%)
015 *--FLAGBYTE (55H) NAAR TAPE
016 *--FILE TYPE BYTE (31H)=MLP NAAR TAPE
017 *
018 *--LENGTE NAAM(2 BYTES) NAAR TAPE + UPD.CHKS.LENGTE
019 *--UPDATED CHKS LENGTEBYTES NAAR TAPE
020 *--NAAM NAAR TAPE + UPD.CHKS NAAM (1340 Hz)
021 *--UPDATED CHKS VAN DE NAAM NAAR TAPE (670 BAUD)
022 *
023 *--LENGTE DADR(2 BYTES) NAAR TAPE + UPD.CHKS.LENGTE
024 *--UPDATED CHKS VAN DADR--LENGTE NAAR TAPE
025 *--DADR NAAR TAPE + UPD.CHKS DADR
026 *--UPDATED CHKS VAN DADR NAAR TAPE
027 *
028 *--LENGTE DATABLOK(2 BYTES) NAAR TAPE+UPD.CHKS.LENGTE
029 *--UPDATED CHKS DATABLOKLENGTE NAAR TAPE
030 *--DATABLOK NAAR TAPE + UPD.CHKS.DATABLOK
031 *--UPDATED CHKS VAN DATABLOK NAAR TAPE
032 *
033 *--AFSLUITSIGNAAL (TRAILER) CA. 2140 Hz.
034 *
035 *INVOEREN m.b.v. SUBST.MEM-TOETS IN SDK 85
036 * :20B0 :NAAM V/H PROGRAMMA )
037 * T/M IN HEXDEC ASCII-CODE )
038 * :20BF )NAME
039 * :20C0 : LOW-ORDER BYTE LENGTE VAN DE NAAM )
040 * :20C1 :HIGH-ORDER BYTE LENGTE VAN DE NAAM )NMLG
041 * :20C2 : LOW-ORDER BYTE LAAGSTE ADRES DATA )
042 * :20C3 :HIGH-ORDER BYTE LAAGSTE ADRES DATA )LADR
043 * :20C4 : LOW-ORDER BYTE BESTEMMINGSADRES DATA )
044 * :20C5 :HIGH-ORDER BYTE BESTEMMINGSADRES DATA )DADR
045 * :20C6 : LOW-ORDER BYTE HOOGSTE ADRES DATA )
046 * :20C7 :HIGH-ORDER BYTE HOOGSTE ADRES DATA )HADR
047 *
048 * OUTPUT: B0 VAN POORT 23H VIA INTERFACE
049 * NAAR DINPLUG.
050 * PROGRAMMA-AANROEP:
051 * ZET TAPERECORDER OF OPNAME !
052 * (GO):0800 (EXEC)
053 *
054 *****
055 NAME EQU :20B0
056 NMLG EQU :20C0
057 LADR EQU :20C2
058 DADR EQU :20C4
059 HADR EQU :20C6
060 TAPSL EQU :3F3F TAPE SPEED LEADER 1:1

```

8080 cassette routines SDK-85

```

002 *****
003 *BRON      :8080 DNA-ASSEMBLER CHR.MTS. ALMELO.
004 *BESTEMMING :SDK 85  6.144 MHz CLOCK (T=326 nSEC)
005 *NAAM      :van OOL
006 *DATUM     :07-06-1983
007 *BUFFERDIMENSIE ASSEMBLER : 12
008 *TELLERSTAND BANDOFPNAME VAN 004 TOT 020
009 *****
010 *TOELICHTING:
011 * HET PROGRAMMA SCHRIJFT DATA NAAR AUDIO-TAPE
012 * VOLGENS DE DAIPc-METHODE, ALS VOLGT:
013 *
014 *--VOORSIGNAAL (LEADER) CA. 1780 Hz (10%)
015 *--FLAGBYTE (55H) NAAR TAPE
016 *--FILE TYPE BYTE (31H)=MLP NAAR TAPE
017 *
018 *--LENGTE NAAM(2 BYTES) NAAR TAPE + UPD.CHKS.LENGTE
019 *--UPDATED CHKS LENGTEBYTES NAAR TAPE
020 *--NAAM NAAR TAPE + UPD.CHKS NAAM (1340 Hz)
021 *--UPDATED CHKS VAN DE NAAM NAAR TAPE (670 BAUD)
022 *
023 *--LENGTE DADR(2 BYTES) NAAR TAPE + UPD.CHKS.LENGTE
024 *--UPDATED CHKS VAN DADR-LENGTE NAAR TAPE
025 *--DADR NAAR TAPE + UPD.CHKS DADR
026 *--UPDATED CHKS VAN DADR NAAR TAPE
027 *
028 *--LENGTE DATABLOK(2 BYTES) NAAR TAPE+UPD.CHKS.LENGTE
029 *--UPDATED CHKS DATABLOKLENGTE NAAR TAPE
030 *--DATABLOK NAAR TAPE + UPD.CHKS.DATABLOK
031 *--UPDATED CHKS VAN DATABLOK NAAR TAPE
032 *
033 *--AFSLUITSIGNAAL (TRAILER) CA. 2140 Hz.
034 *
035 *INVOEREN m.b.v. SUBST.MEM-TOETS IN SDK 85
036 * :20B0 :NAAM V/H PROGRAMMA )
037 * T/M IN HEXDEC ASCII-CODE )
038 * :20BF )NAME
039 * :20C0 : LOW-ORDER BYTE LENGTE VAN DE NAAM )
040 * :20C1 :HIGH-ORDER BYTE LENGTE VAN DE NAAM )NMLG
041 * :20C2 : LOW-ORDER BYTE LAAGSTE ADRES DATA )
042 * :20C3 :HIGH-ORDER BYTE LAAGSTE ADRES DATA )LADR
043 * :20C4 : LOW-ORDER BYTE BESTEMMINGSADRES DATA )
044 * :20C5 :HIGH-ORDER BYTE BESTEMMINGSADRES DATA )DADR
045 * :20C6 : LOW-ORDER BYTE HOOGSTE ADRES DATA )
046 * :20C7 :HIGH-ORDER BYTE HOOGSTE ADRES DATA )HADR
047 *
048 * OUTPUT: B0 VAN POORT 23H VIA INTERFACE
049 * NAAR DINPLUG.
050 * PROGRAMMA-AANROEP:
051 * ZET TAPERECORDER OF OFNAME !
052 * (GO):0800 (EXEC)
053 *
054 *****
055 NAME EQU :20B0
056 NMLG EQU :20C0
057 LADR EQU :20C2
058 DADR EQU :20C4
059 HADR EQU :20C6
060 TAPSL EQU :3F3F TAPE SPEED LEADER 1:1

```

8080 cassette routines SDK-85

061	TAPSD	EQU	:693F	TAPE SPEED DATA	5:3
062	TAPST	EQU	:2A3F	TAPE SPEED TRAILER	2:3
063	FUNCT1	EQU	:0921	ROUTINE IN CASSrd	
064	FUNCT2	EQU	:0939	ROUTINE IN CASSrd	
065		ORG	:0800	BEGINADRES IN EPROM SDK 85	
066	0800 F3	CASSrc	DI	BLOKKEER ALLE INTERRUPTS	
067	0801 31B020		LXI SP,:20B0	INIT STACKPNT	
068	0804 CD2608		CALL WROPEN	OPENINGSROUTINE	
069	0807 110200		LXI D,:0002	DADR IS 2 BYTES LANG	
070	080A 21C420		LXI H,DADR		
071	080D CD4A08		CALL WRBLOK	DADR NAAR TAPE	
072	0810 2AC620		LHLD HADR	HADR IN (HL)	
073	0813 EB		XCHG	HADR IN (DE)	
074	0814 2AC220		LHLD LADR	LADR IN (HL)	
075	0817 13		INX D	HADR+1	
076	0818 7B		MOV A,E	DATABLOK=(HADR+1)-LADR	
077	0819 95		SUB L)	
078	081A 5F		MOV E,A)	
079	081B 7A		MOV A,D)	
080	081C 9C		SEB H)	
081	081D 57		MOV D,A) (DE)=(DE)-(HL)	
082	081E CD4A08		CALL WRBLOK	DATABLOK NAAR TAPE	
083	0821 CDE908		CALL TRAILR	AFSLUITROUTINE	
084	0824 FB		EI	INTERRUPTS TOEGESTAAN	
085	0825 C7		RST 0	KLAAR: DISPL - 80 85	
086	0826 3E0F	WROPEN	MVI A,:0F	INITCODE POORTEN	
087	0828 D320		OUT :20	NAAR CSR:POORT 23H IS OUTP.	
088	082A 32FF20		STA :20FF	INITCODE NAAR STACK	
089	082D CD2109		CALL FUNCT1	CASS IN ADRESVELD	
090	0830 CD3909		CALL FUNCT2	rc IN DATAVELD	
091	0833 CD9008		CALL LEADER	VOORSIGNAAL	
092	0836 3E55		MVI A,:55	FLAGBYTE	
093	0838 CDA608		CALL WRBYTE	NAAR TAPE	
094	083B 3E31		MVI A,:31	FILE TYPE BYTE (ASCII=1)	
095	083D CDA608		CALL WRBYTE	NAAR TAPE	
096	0840 2AC020		LHLD NMLG	NMLG-ADRES	
097	0843 EB		XCHG	NMLG IN (DE)	
098	0844 CD7808		CALL LGTST	TEST LENGTE V/D NAAM	
099	0847 21B020		LXI H,NAME	PNT NAAR BEGINADRES NAAM	
100	084A CD5D08	WRBLOK	CALL WRPNT	POINTERS NAAR TAPE	
101	084D 0656		MVI B,:56	INIT CHECKSUM	
102	084F 7A	WBLOK	MOV A,D)	
103	0850 B3		ORA E)BLOKTELLER	
104	0851 CA7308		JZ WRCHKS	BLOK=0? CHKS NAAR TAPE	
105	0854 1B		DCX D	VERLAAG BLOKTELLER	
106	0855 7E		MOV A,M	DATA NAAR ACCU	
107	0856 23		INX H	VERHOOG RAM-POINTER	
108	0857 CD6C08		CALL WDUFDC	DATA NAAR TAPE + UPDATE CHKS	
109	085A C34F08		JMP WBLOK	VOLGENDE DATABYTE	
110	085D 0656	WRPNT	MVI B,:56	INIT CHECKSUM	
111	085F 7A		MOV A,D	HIGH-ORDER-LENGTEBYTE	
112	0860 CD6C08		CALL WDUFDC	NAAR TAPE + UPD CHKS	
113	0863 7B		MOV A,E	LOW-ORDER-LENGTEBYTE	
114	0864 CD6C08		CALL WDUFDC	NAAR TAPE + UPD CHKS	
115	0867 7B		MOV A,B	CHECKSUM BLOKLENGTE	
116	0868 CDA608		CALL WRBYTE	NAAR TAPE	
117	086B C9		RET		
118	086C CDA608	WDUFDC	CALL WRBYTE	DATABYTE NAAR TAPE	
119	086F AB		XRA B	BEPAAL CHKS DATABYTE	
120	0870 07		RLC	ROTEER CHKS	

```

181 08D4 F1          POP   PSW
182 08D5 C9          RET
183 08D6 3E01        OUTPUT MVI  A,:01      BO="1"
184 08D8 D323        OUT   :23      START POS.PERIODENTIJD
185 08DA 25          DEL1  DCR  H        VERLAAG TAPSL/D/T
186 08DB C2DA08      JNZ   DEL1      WACHT HALVE PERIODETIJD
187 08DE 2D          DCR  L
188 08DF 2D          DCR  L
189 08E0 2D          DCR  L        (SYMMETRIE)
190 08E1 3D          DCR  A        BO="0"
191 08E2 D323        OUT   :23      START NEG.PERIODENTIJD
192 08E4 2D          DEL2  DCR  L        VERLAAG TAPSL/D/T
193 08E5 C2E408      JNZ   DEL2      WACHT HALVE PERIODENTIJD
194 08E8 C9          RET
195 08E9 213F2A      TRAILR LXI  H,TAPST  SIGNAALVERHOUDING 2:3
196 08EC 010001      LXI  B,:0100  TIJDSDUUR TRLR
197 08EF C39608      JMP  TONE      GENEREER TRAILER
198 08F2          END   END

```

```

*****
* S Y M B O L   T A B L E *
*****

```

```

CASSrc 0800   DADR  20C4   DEL1  08DA   DEL2  08E4
END      08F2   ERROR 0884   FUNCT1 0921  FUNCT2 0939
HADR    20C6   LADR  20C2   LEADER 0890  LGTST  0878
NAME    20B0   NMLG  20C0   OUTPUT 08D6  TAPSD  693F
TAPSL   3F3F   TAPST 2A3F   TONE   0896  TRAILR 08E9
WBLOK   084F   WBYTE 08B1   WDUFDC 086C  WRBIT  08C3
WRBLOK  084A   WRBYTE 08A6  WRCHKS 0873  WROPEN 0826
WRPNT   085D

```

```

0800 F3 31 B0 20 CD 26 08 11 02 00 21 C4 20 CD 4A 08
0810 2A C6 20 EB 2A C2 20 13 7B 95 5F 7A 9C 57 CD 4A
0820 08 CD E9 08 FB C7 3E 0F D3 20 32 FF 20 CD 21 09
0830 CD 39 09 CD 90 08 3E 55 CD A6 08 3E 31 CD A6 08
0840 2A C0 20 EB CD 78 08 21 B0 20 CD 5D 08 06 56 7A
0850 B3 CA 73 08 1B 7E 23 CD 6C 08 C3 4F 08 06 56 7A
0860 CD 6C 08 7B CD 6C 08 78 CD A6 08 C9 CD A6 08 A8
0870 07 47 C9 78 CD A6 08 C9 7A B7 C2 84 08 7B FE 11
0880 D2 84 08 C9 AF 47 21 9E 03 CD B7 02 FB C3 66 00
0890 21 3F 3F 01 EB 07 CD C3 08 0B 78 B1 C2 96 08 21
08A0 3F 69 CD C3 08 C9 F5 C5 D5 E5 21 3F 69 5C 55 06
08B0 08 17 DC C3 08 EB D4 C3 08 EB 05 C2 B1 08 E1 D1
08C0 C1 F1 C9 F5 E5 6C CD D6 08 E1 E5 65 7D D6 08 6F
08D0 CD D6 08 E1 F1 C9 3E 01 D3 23 25 C2 DA 08 2D 2D
08E0 2D 3D D3 23 2D C2 E4 08 C9 21 3F 2A 01 00 01 C3
08F0 96 08 20

```

```

002 *****
003 *BRON      : 8080 DNA-ASSEMBLER CHR.MTS ALMELO
004 *BESTEMMING : SDK85 6.144 MHz CLOCK(T=326 nSEC)
005 *NAAM      : van OOL
006 *DATUM     : 07-06-1983
007 *TELLERSTAND BANDOPNAME VAN 050 TOT 075
008 *BUFFERDIMENSIE ASSEMBLER : 12
009 *****
010 *TOELICHTING:
011 * HET PROGRAMMA LEEST DATA VAN AUDIO-TAPE
012 * VOLGENS DE DAIPc-METHODE. VERDER WORDEN
013 * ONDERSTAANDE POINTERS IN HET GEHEUGEN
014 * GEPLAATST, ZODAT EEN GELEZEN PROGRAMMA
015 * ZONDER VEEL VOORBEREIDING OOK WEER KAN
016 * WORDEN OPGENOMEN M.B.V. CASSrc V1.0 .
017 * HET PROGRAMMA TEST NIET OF HET TE LADEN
018 * PROGRAMMA PAST IN DE BESCHIKBARE
019 * GEHEUGENRUIMTE VAN DE SDK 85 !
020 * N.B.: DE OFFSET MOET VOOR HET STARTEN
021 * VAN HET CASSrd-PROGRAMMA WORDEN
022 * INGEVORDERD. HET IN TE LEZEN PROGRAMMA
023 * WORDT OP EEN PLAATS IN HET
024 * GEHEUGEN(LADR) GEZET, WELKE GELIJK IS
025 * AAN (DADR+OFFSET)
026 * DE OFFSET MAAKT HET MOGELIJK EEN PROGRAMMA,
027 * INDIEN NOODZAKELIJK, OP EEN ANDER ADRES
028 * IN TE LEZEN, DAN OP DE CASSETTE-
029 * BAND STAAT. (LADR=DADR+OFFS)
030 * INDIEN OFFS=0000,DAN IS (LADR=DADR)
031 *
032 * :20B0 :NAAM V/H PROGRAMMA )
033 * T/M IN HEXDEC ASCII-CODE )
034 * :20BF )NAME
035 * :20C0 : LOW-ORDER BYTE LENGTE V/D NAAM )
036 * :20C1 :HIGH-ORDER BYTE LENGTE V/D NAAM )NMLG
037 * :20C2 : LOW-ORDER BYTE LAAGSTE ADRES DATA )
038 * :20C3 :HIGH-ORDER BYTE LAAGSTE ADRES DATA )LADR
039 * :20C4 : LOW-ORDER BYTE BESTEMMINGSadRES )
040 * :20C5 :HIGH-ORDER BYTE BESTEMMINGSadRES )DADR
041 * :20C6 : LOW-ORDER BYTE HOOGSTE ADRES DATA )
042 * :20C7 :HIGH-ORDER BYTE HOOGSTE ADRES DATA )HADR
043 * :20C8 : LOW-ORDER BYTE OFFSET )
044 * :20C9 :HIGH-ORDER BYTE OFFSET )OFFS
045 *
046 * INPUT: B1 VAN POORT 23H VIA INTERFACE
047 * NAAR DINPLUG
048 * PROGRAMMA-AANROEP:
049 * (GO): 0900 (EXEC)
050 * START DAARNA DE CASSETTERECORDER AAN HET
051 * BEGIN VAN HET TE LADEN PROGRAMMA.
052 *****
053 NAME EQU :20B0 NAAMADRES
054 NMLG EQU :20C0
055 LADR EQU :20C2
056 DADR EQU :20C4
057 HADR EQU :20C6
058 OFFS EQU :20C8
059 ERROR EQU :0884 ROUTINE IN CASSrc
060 LGTST EQU :0878 IDEM

```

061		ORG	:0900	BEGINADRES IN EPROM SDK 85
062	0900 F3	CASSrd	DI	GEEN INTERRUPTS TOEGESTAAN
063	0901 31B020		LXI SP,:20B0	INIT STACKPNT
064	0904 CD5909		CALL RDOPEN	OPENINGSROUTINE
065	0907 21C420		LXI H,DADR	(HL)=DADR-ADRES
066	090A CD8D09		CALL RDBLOK	LEES DADR VAN TAPE
067	090D 2AC420		LHLD DADR	(HL)=DADR
068	0910 EB		XCHG	DADR IN (DE)
069	0911 2AC820		LHLD OFFS	OFFSET IN (HL)
070	0914 19		DAD D	LADR=DADR+OFFSET
071	0915 22C220		SHLD LADR	LADR-ADRES IN SDK 85 RAM
072	0918 CD8D09		CALL RDBLOK	LEES DATABLOK VAN TAPE
073	091B 2B		DCX H	(HL)=HADR
074	091C 22C620		SHLD HADR	HADR IN RAM
075	091F FB		EI	INTERRUPTS TOEGESTAAN
076	0920 C7		RST 0	KLAAR: DISPL - 80 85
077	0921 3E90	FUNCT1	MVI A,:90	ADRESVELDCODE
078	0923 320019		STA :1900	NAAR COMMANDREG.
079	0926 3E6C		MVI A,:6C	C
080	0928 320018		STA :1800	NAAR DISPLAY
081	092B 3E88		MVI A,:88	A
082	092D 320118		STA :1801	IDEM
083	0930 3E29		MVI A,:29	S
084	0932 320218		STA :1802	
085	0935 320318		STA :1803	S
086	0938 C9		RET	
087	0939 3E94	FUNCT2	MVI A,:94	DATAVELDCODE
088	093B 320419		STA :1904	NAAR COMMANDREG.
089	093E 3EFA		MVI A,:FA	r
090	0940 320418		STA :1804	NAAR DISPLAY
091	0943 3E7A		MVI A,:7A	c
092	0945 320518		STA :1805	NAAR DISPLAY
093	0948 C9		RET	
094	0949 3E94	FUNCT3	MVI A,:94	DATAVELDCODE
095	094B 320419		STA :1904	NAAR COMMANDREG.
096	094E 3EFA		MVI A,:FA	r
097	0950 320418		STA :1804	NAAR DISPLAY
098	0953 3E1A		MVI A,:1A	d
099	0955 320518		STA :1805	
100	0958 C9		RET	
101	0959 CD2109	RDOPEN	CALL FUNCT1	CASS IN DISPLAY
102	095C CD4909		CALL FUNCT3	rd IN DISPLAY
103	095F 3E03		MVI A,:03	INIT-CODE VOOR POORTEN
104	0961 D320		OUT :20	POORT 23H IS INPUT
105	0963 32FF20		STA :20FF	INITCODE NAAR STACK
106	0966 CDCF09	START	CALL RDLEAD	LEES VOORSIGNAAL VAN TAPE
107	0969 CD230A		CALL RDBYTE	LEES FLAGBYTE 55H VAN TAPE
108	096C DA6609		JC START	CY=LEESFOUT
109	096F FE55		CPI :55	TEST FLAGBYTE
110	0971 C26609		JNZ START	TEST NIET OK?DAN OPNIEUW
111	0974 CD230A		CALL RDBYTE	LEES FILE-TYPE BYTE VAN TAPE
112	0977 DA6609		JC START	CY=LEESFOUT
113	097A FE31		CPI :31	TEST FILE-TYPE
114	097C C26609		JNZ START	31H=ASCII "1" = MLP
115	097F 21B020		LXI H,NAME	ADRES PRGR-NAAM
116	0982 CD8D09		CALL RDBLOK	LEES NAAM VAN TAPE
117	0985 CD7808		CALL LGTST	TEST DE LENGTE V/D NAAM
118	0988 EB		XCHG	LENGTE NAAM IN (HL)
119	0989 22C020		SHLD NMLG	NMLG IN RAM
120	098C C9		RET	

121	098D	CDAB09	RDBLOK	CALL	RDPNT	LEES LENGTEPNT VAN TAPE
122	0990	D5		PUSH	D	BEWAAR LENGTEPOINTER
123	0991	DA8408		JC	ERROR	CY = LEESFOUT
124	0994	B7		ORA	A	TEST CHKS
125	0995	C28408		JNZ	ERROR	CHKS FOUT
126	0998	0656		MVI	B, :56	INIT CHKS
127	099A	7A	RBLOK	MOV	A, D)
128	099B	B3		ORA	E)DATABLOKTELLER
129	099C	CAC309		JZ	RDCHKS	DATABL.=0?LEES CHKS
130	099F	1B		DCX	D	VERLAAG BLOKTELLER
131	09A0	CDBA09		CALL	RDUPDC	LEES DATA+UPD CHKS
132	09A3	DA8408		JC	ERROR	CY=LEESFOUT
133	09A6	77		MOV	M, A	ZET DATA IN RAM
134	09A7	23		INX	H	VERHOOG RAMPNT
135	09A8	C39A09		JMP	RBLOK	LEES VOLG. DATABYTE
136	09AB	0656	RDPNT	MVI	B, :56	INIT CHKS
137	09AD	CDBA09		CALL	RDUPDC	LEES DATA+UPD CHKS
138	09B0	57		MOV	D, A	HIGH-ORDER BYTE LENGTE
139	09B1	D4BA09		CNC	RDUPDC	LEES DATA+UPD CHKS
140	09B4	5F		MOV	E, A	LOW-ORDER BYTE LENGTE
141	09B5	D4230A		CNC	RDBYTE	LEES CHKS LENGTE
142	09B8	90		SUB	B	VERGELIJK CHKS EN UPD-CHKS
143	09B9	C9		RET		
144	09BA	CD230A	RDUPDC	CALL	RDBYTE	LEES DATA VAN TAPE
145	09BD	F5		PUSH	PSW	BEWAAR DATABYTE
146	09BE	A8		XRA	B	BEPAAL CHKS
147	09BF	07		RLC		ROTEER CHKS
148	09C0	47		MOV	B, A	UPDATED CHKS IN B
149	09C1	F1		POP	PSW	HERSTEL DATABYTE
150	09C2	C9		RET		
151	09C3	CD230A	RDCHKS	CALL	RDBYTE	LEES CHKS VAN TAPE
152	09C6	DA8408		JC	ERROR	CY=LEESFOUT
153	09C9	BB		CMP	B	VERGELIJK CHKS EN UPDCHKS
154	09CA	C28408		JNZ	ERROR	CHKS FOUT, DAN FOUTMELDING
155	09CD	D1		POP	D	HERSTEL LENGTEPOINTER
156	09CE	C9		RET		
157	09CF	062B	RDLEAD	MVI	B, 40	GESCHATTE PULSDUUR
158	09D1	DB23	NOSIGN	IN	:23	LEES POORT 23H
159	09D3	E602		ANI	:02	SELECTEER B1
160	09D5	C2D109		JNZ	NOSIGN	WACHT OP LDRSIGN
161	09D8	4B		MOV	C, B	GESCHATTE/WERK. PULSDUUR
162	09D9	1614		MVI	D, 20	MINIMAAL AANTAL LDRPULSEN
163	09DB	1E00	CYCLE	MVI	E, 0	LEESCYCLUS
164	09DD	1D	DELAY2	DCR	E	PULSDUURTELLER
165	09DE	CAD109		JZ	NOSIGN	GEEN SIGN.?DAN OPNIEUW
166	09E1	DB23		IN	:23	LEES POORT 23H
167	09E3	E602		ANI	:02	SELECTEER B1
168	09E5	CADD09		JZ	DELAY2	B1="0", DAN WACHT
169	09E8	0600		MVI	B, 0	INIT LDRCHECK
170	09EA	04	LDRCHK	INR	B	(B)=GEMETEN PULSDUUR
171	09EB	CAD109		JZ	NOSIGN	GEEN SIGN.?DAN OPNIEUW
172	09EE	DB23		IN	:23	LEES INPUTPOORT
173	09F0	E602		ANI	:02	SELECTEER B1
174	09F2	00		NOP		
175	09F3	C2EA09		JNZ	LDRCHK	WACHT OP LAAG SIGNAAL
176	09F6	7B		MOV	A, B	PULSDUUR IN (A)
177	09F7	91		SUB	C	BEREKEN VERSCHIL PULSDUUR
178	09F8	F2FD09		JF	TEST%	PULSDUUR>STANDAARD?
179	09FB	2F		CMA		PULSDUUR<STANDAARD? DAN
180	09FC	3C		INR	A	BEPAAL POS.WAARDE

181	09FD	5F	TEST%	MOV	E,A	POS.WAARDE VERSCHIL IN (E)
182	09FE	79		MOV	A,C	STANDAARD PULSDUUR IN (A)
183	09FF	E6F0		ANI	:F0)
184	0A01	1F		RAR)BEREKENING VAN TOEGESTANE
185	0A02	1F		RAR)AFWIJKING (10%)
186	0A03	1F		RAR) (A)=04 IS 10% VAN 40 (C)
187	0A04	BB		CMP	E	VERSCHIL =TOEGESTANE ?
188	0A05	DA100A		JC	NOSYNC	VERSCHIL>10%,DAN NOSYNC
189	0A08	15		DCR	D	LDRPULSTELLER
190	0A09	C2DB09		JNZ	CYCLE	HERHAAL CYCLUS TOTDAT (D)=0
191	0A0C	14		INR	D	BIJ SYNC: (D)=01
192	0A0D	C3DB09		JMP	CYCLE	GA DOOR TOT ASYNCHRONE PULS
193	0A10	15	NOSYNC	DCR	D	BIJ SYNC: (D)=00
194	0A11	C2D109		JNZ	NOSIGN	GEEN SYNC:DAN OPNIEUW
195	0A14	DB23	LOW	IN	:23	LEES ASYNCHRONE LDRPULS
196	0A16	E602		ANI	:02	SELECTEER B1
197	0A18	CA140A		JZ	LOW	WACHT OP HOOG SIGNAAL
198	0A1B	DB23	HIGH	IN	:23	LEES ASYNCHRONE LDRPULS
199	0A1D	E602		ANI	:02	SELECTEER B1
200	0A1F	C21B0A		JNZ	HIGH	WACHT OP LAAG SIGNAAL
201	0A22	C9		RET		
202	0A23	C5	RDBYTE	PUSH	B	
203	0A24	D5		PUSH	D	
204	0A25	E5		PUSH	H	
205	0A26	1EFE		MVI	E,:FE	BYTETELLER
206	0A28	CD390A	RBYTE	CALL	RDBIT	LEES BIT VANTAPE
207	0A2B	DA350A		JC	POPRET	CY=LEESFOOT
208	0A2E	17		RAL		ROTEER BITWAARDE(B7) IN CY
209	0A2F	7B		MOV	A,E	BYTETELLER IN ACCU
210	0A30	17		RAL		BYTETELLER+BIT(S) IN (A)
211	0A31	5F		MOV	E,A	BEWAAR BYTETELLER +BIT(S)
212	0A32	DA280A		JC	RBYTE	VOLGENDE BIT.8 BITS=BYTE
213	0A35	E1	POPRET	POP	H	
214	0A36	D1		POP	D	
215	0A37	C1		POP	B	
216	0A38	C9		RET		
217	0A39	AF	RDBIT	XRA	A	RESET A,B,C,D EN CY
218	0A3A	57		MOV	D,A	INIT PERIODE-
219	0A3B	47		MOV	B,A	TIJDENDELAYS EN
220	0A3C	4F		MOV	C,A	BITWAARDETELLER IN (D)
221	0A3D	05	NEG1	DCR	B	BEGIN NEG.PERIODENTIJD
222	0A3E	CA700A		JZ	SETCY	SIGNAALONDERBREKING?=CY
223	0A41	DB23		IN	:23	LEES POORT 23H
224	0A43	E602		ANI	:02	SELECTEER B1
225	0A45	CA3D0A		JZ	NEG1	NOG STEEDS NEG.?
226	0A48	0D	POS1	DCR	C	BEGIN POS.PERIODENTIJD
227	0A49	CA700A		JZ	SETCY	SIGNAALONDERBREKING?=CY
228	0A4C	15		DCR	D	VERLAAG BITWAARDETELLER(BW1)
229	0A4D	DB23		IN	:23	LEES POORT 23H
230	0A4F	E602		ANI	:02	SELECTEER B1
231	0A51	C2480A		JNZ	POS1	NOG STEEDS POS1?
232	0A54	010000		LXI	B,:0000	RESET (BC)
233	0A57	05	NEG2	DCR	B	BEGIN NEG.PERIODENTIJD
234	0A58	CA700A		JZ	SETCY	SIGNAALONDRBREKING?=CY
235	0A5B	DB23		IN	:23	LEES POORT 23H
236	0A5D	E602		ANI	:02	SELECTEER B1
237	0A5F	CA570A		JZ	NEG2	NOG STEEDS NEGATIEF?
238	0A62	0D	POS2	DCR	C	BEGIN POS.PERIODENTIJD
239	0A63	CA700A		JZ	SETCY	SIGNAALONDERBREKING?=CY
240	0A66	14		INR	D	VERHOOG BITWAARDETELLER(BW2)

```

241 0A67 DB23          IN      :23      LEES POORT 23H
242 0A69 E602          ANI      :02      SELECTEER B1
243 0A6B C2620A        JNZ      POS2      NOG STEEDS POS2?
244 0A6E 7A            MOV      A,D        BW1>BW2:BITWAARDE="1" (B7)
245 0A6F C9            RET                          BW1<BW2:BITWAARDE="0" (B7)
246 0A70 37            SETCY    STC        ZET CY VOOR FOUTMELDING
247 0A71 C9            RET
248 0A72                END      END
    
```

* S Y M B O L T A B L E *

```

CASSrd 0900    CYCLE 09DB    DADR    20C4    DELAY2 09DD
END    0A72    ERROR 08B4    FUNCT1 0921    FUNCT2 0939
FUNCT3 0949    HADR    20C6    HIGH    0A1B    LADR    20C2
LDRCHK 09EA    LGTST 0878    LOW    0A14    NAME    20B0
NEG1    0A3D    NEG2    0A57    NMLG    20C0    NOSIGN 09D1
NOSYNC 0A10    OFFS    20C8    POPRET 0A35    POS1    0A48
POS2    0A62    RBLOK 099A    RBYTE    0A28    RDBIT    0A39
RDBLOK 098D    RDBYTE 0A23    RDCHKS 09C3    RDLEAD 09CF
RDOPEN 0959    RDPNT 09AB    RDUPDC 098A    SETCY    0A70
START    0966    TEST% 09FD
    
```

```

0900 F3 31 B0 20 CD 59 09 21 C4 20 CD 8D 09 2A C4 20
0910 EB 2A C8 20 19 22 C2 20 CD 8D 09 2B 22 C6 20 FB
0920 C7 3E 90 32 00 19 3E 6C 32 00 18 3E 88 32 01 18
0930 3E 29 32 02 18 32 03 18 C9 3E 94 32 04 19 3E FA
0940 32 04 18 3E 7A 32 05 18 C9 3E 94 32 04 19 3E FA
0950 32 04 18 3E 1A 32 05 18 C9 CD 21 09 CD 49 09 3E
0960 03 D3 20 32 FF 20 CD CF 09 CD 23 0A DA 66 09 FE
0970 55 C2 66 09 CD 23 0A DA 66 09 FE 31 C2 66 09 21
0980 B0 20 CD 8D 09 CD 78 08 EB 22 C0 20 C9 CD AB 09
0990 D5 DA 84 08 B7 C2 84 08 06 56 7A B3 CA C3 09 1B
09A0 CD BA 09 DA 84 08 77 23 C3 9A 09 06 56 CD BA 09
09B0 57 D4 BA 09 5F D4 23 0A 90 C9 CD 23 0A F5 AB 07
09C0 47 F1 C9 CD 23 0A DA 84 08 B8 C2 84 08 D1 C9 06
09D0 2B DB 23 E6 02 C2 D1 09 48 16 14 1E 00 1D CA D1
09E0 09 DB 23 E6 02 CA DD 09 06 00 04 CA D1 09 DB 23
09F0 E6 02 00 C2 EA 09 78 91 F2 FD 09 2F 3C 5F 79 E6
0A00 F0 1F 1F 1F BB DA 10 0A 15 C2 DB 09 14 C3 DB 09
0A10 15 C2 D1 09 DB 23 E6 02 CA 14 0A DB 23 E6 02 C2
0A20 1B 0A C9 C5 D5 E5 1E FE CD 39 0A DA 35 0A 17 7B
0A30 17 5F DA 28 0A E1 D1 C1 C9 AF 57 47 4F 05 CA 70
0A40 0A DB 23 E6 02 CA 3D 0A 0D CA 70 0A 15 DB 23 E6
0A50 02 C2 48 0A 01 00 00 05 CA 70 0A DB 23 E6 02 CA
0A60 57 0A 0D CA 70 0A 14 DB 23 E6 02 C2 62 0A 7A C9
0A70 37 C9 00
    
```

Programmes mathématiques

```

10 PRINT CHR$(12):LIST 20-240:CALLM#D6DA
20 REM #####
30 REM ##### PROGRAMME ECRIT ET REALISE #####
40 REM ##### PAR #####
50 REM ##### #####
60 REM ##### Fabrice DULUINS #####
70 REM ##### #####
80 REM #####
90 REM ##### N.B : #####
100 REM ##### #####
110 REM ##### JE CHERCHE CORRESPONDANTS #####
120 REM ##### DE TOUT PAYS EN VUE #####
130 REM ##### D'ECHANGES DE PROGRAMMES #####
140 REM ##### INEDITS #####
150 REM ##### #####
160 REM ##### ECRIRE A : #####
170 REM ##### #####
180 REM ##### Fabrice DULUINS #####
190 REM ##### 4 allée de la Tour Renard #####
200 REM ##### B - 1400 NIVELLES #####
210 REM ##### BELGIQUE #####
220 REM ##### #####
230 REM ##### SPACEBAR #####
240 REM #####
250 MODE 0:PRINT CHR$(12.0):PRINT "PRODUITS REMARQUABLES
1"
260 PRINT "FORMULES DE SIMPSON 2"
270 PRINT "MOYENNE ARITHMETIQUE 3"
280 PRINT "CERCLE DETERMINE PAR TROIS POINTS 4"
290 PRINT "HARMONIC NUMBER 5"
300 PRINT "LOGARITHME SUR N' IMPORTE QUELLE BASE 6"
310 PRINT "EQUATION DU SECOND DEGRE 7"
320 PRINT "SYSTEME D'EQUATIONS A 2 INCONNUES 8"
330 PRINT "NOMBRE PREMIER ET PLUS PETIT FACTEUR 9"
340 PRINT "ALL GRAPHS 10"
350 PRINT :PRINT "STOP HALT ":INPUT "VOTRE
CHOIX ";A$
360 PRINT CHR$(12):IF A$="1" GOTO 480
370 IF A$="2" GOTO 580
380 IF A$="3" GOTO 710
390 IF A$="4" GOTO 830
400 IF A$="5" GOTO 1020
410 IF A$="6" GOTO 1180
420 IF A$="7" GOTO 1300
430 IF A$="8" GOTO 1560
440 IF A$="9" GOTO 1740
450 IF A$="10" GOTO 1960
460 IF A$="HALT" THEN 470
470 END
480 PRINT CHR$(12):PRINT " PRODUITS REMARQUABLES"
490 PRINT :PRINT "(A+B)^2 = A^2 + 2*A*B + B^2"
500 FOR I=1.0 TO NB
510 PRINT :PRINT "(A-B)^2 = A^2 - 2*A*B +B^2"
520 PRINT :PRINT "(A+B)*(A-B) = A^2 -B^2"
530 PRINT :PRINT "(A+B)^3 = A^3 + 3*A^2*B + 3*A*B^2 + B^3"
540 PRINT :PRINT "(A-B)^3 = A^3 - 3*A^2*B + 3*A*B^2 - B^3"
550 PRINT :PRINT "(A-B)*(A^2 + A*B + B^2) = A^3 - B^3"
560 PRINT :PRINT "(A+B)*(A^2 - A*B + B^2) = A^3 + B^3"
570 WAIT TIME 150:GOTO 250
580 PRINT " FORMULES DE SIMPSON "
590 PRINT :PRINT "COS P + COS Q = 2 * COS (P+Q)/2 * COS (P-Q)/2"
600 PRINT :PRINT "COS P - COS Q = 2 * COS (P+Q)/2 * COS (P-Q)/2"
610 PRINT :PRINT "SIN P - SIN Q = 2 * SIN (P+Q)/2 * COS (P-Q)/2"
620 PRINT :PRINT "SIN P + SIN Q = 2 * COS (P+Q)/2 * SIN (P-Q)/2"
630 PRINT :PRINT "COS (A-B) = COS A * COS B + SIN A * SIN B"

```



```

640 PRINT :PRINT "COS (A+B) = COS A * COS B - SIN A * SIN B"
650 PRINT :PRINT "SIN (A+B) = SIN A * COS B + COS A * SIN B"
660 PRINT :PRINT "SIN (A-B) = SIN A * COS B - COS A * SIN B"
670 PRINT :PRINT "TG (A+-B) = (TG A +- TG B)/(1 +- TG A * TG B)"
680 PRINT :PRINT "SIN 2*A = 2 * SIN A * COS A"
690 PRINT :PRINT "COS 2*A = COS^2 A - SIN^2 A = 1 - 2 *SIN^2 A = 2 * COS^
2 - 1"
700 WAIT TIME 150:GOTO 250
710 PRINT "          MOYENNE ARITHMETIQUE "
720 LET X=0.0
730 LET N=0.0
740 PRINT :PRINT "ENTREZ LE NOMBRE ";
750 INPUT W
760 IF W=0.0 GOTO 250
770 LET N=N+1.0
780 LET X=X+W
790 LET A=X/N
800 PRINT :PRINT " N= ";N;" LE NOMBRE EST ";W
810 PRINT "LA MOYENNE EST DE ";A
820 GOTO 740
830 PRINT "          CERCLE DETERMINE PAR TROIS POINTS"
840 PRINT :INPUT "COORD DU 1e PT =";X1,Y1
850 PRINT :INPUT "COORD DU 2e PT =";X2,Y2
860 PRINT :INPUT "COORD DU 3e PT =";X3,Y3
870 LET A=(Y2-Y1)/(X2-X1)
880 LET B=(Y3-Y1)/(X3-X1)
890 LET C=((X2-X1)*(X2+X1))+((Y2-Y1)*(Y2+Y1))
900 LET D=C/(2.0*(X2-X1))
910 LET E=((X3-X1)*(X3+X1))+((Y3-Y1)*(Y3+Y1))
920 LET F=E/(2.0*(X3-X1))
930 LET Y0=(F-D)/(B-A)
940 LET X0=F-(B*Y0)
950 LET R=SQR((ABS(X3-X0))^2.0+(ABS(Y3-Y0))^2.0)
960 PRINT :PRINT "COORD DU CENTRE X0,Y0 = ";X0;" , ";Y0
970 PRINT "RAYON = ";R
980 PRINT :PRINT "POUR CONTINUER TAPEZ SUR 1,SINON SUR 0"
990 INPUT L
1000 IF L=1.0 THEN GOTO 840
1010 GOTO 250
1020 PRINT "          HARMONIC NUMBERS"
1030 PRINT :PRINT "NOMBRE MAXIMUM DE TERMES"
1040 INPUT N
1050 LET K=0.0
1060 LET D=0.0
1070 PRINT :PRINT "NOMBRE DE TERMES ,VALEUR"
1080 FOR I=0.0 TO N
1090 LET K=I+1.0
1100 LET C=1.0/K
1110 LET D=D+C
1120 PRINT K,D
1130 NEXT I:PRINT
1140 PRINT "POUR CONTINUER TAPEZ SUR 1,SINON SUR 0"
1150 INPUT L
1160 IF L=1.0 THEN GOTO 1020
1170 GOTO 250
1180 PRINT :PRINT "          LOG OF ANY BASE"
1190 REM BASE Y
1200 PRINT :INPUT " BASE = ";Y
1210 PRINT :INPUT " X = ";X
1220 J=LOG(X)/LOG(Y)
1230 PRINT :PRINT " LOG ";X;" EN BASE ";Y;" = ";J
1240 PRINT :PRINT "POUR CONTINUER TAPEZ SUR 1,SINON SUR 0"
1250 INPUT L
1260 IF L=1.0 THEN 1280
1270 GOTO 250
1280 PRINT

```

```

1290 GOTO 1200
1300 PRINT "          EQUATION DU SECOND DEGRE"
1310 PRINT :PRINT " AX^2 + BX + C = 0 "
1320 PRINT :INPUT "ENTREZ LES VALEURS DE A, B, ET C ";A,B,C
1330 D=((ABS(B)^2.0)-(4.0*A*C))/(4.0*ABS(A)^2.0):PRINT :PRINT " D= ";D
1340 IF D>=0.0 THEN 1410
1350 X=B/(2.0*A)
1360 Y=(SQR((4.0*A*C)-ABS(B)^2.0))/(2.0*A)
1370 PRINT :PRINT "Reponses complexes"
1380 PRINT "Part reelle      = ";X
1390 PRINT "Part imaginaire = ";Y
1400 GOTO 1500
1410 E=-B/(2.0*A)
1420 IF E>=0.0 THEN 1450
1430 Z=E-SQR(D)
1440 GOTO 1460
1450 Z=E+SQR(D)
1460 W=C/(Z*A)
1470 PRINT :PRINT "Reponses reelles"
1480 PRINT "1e reponse = ";Z
1490 PRINT "2e reponse = ";W
1500 PRINT "*****"
1510 INPUT "POUR CONTINUER SUR 1, SINON SUR 0 ";L
1520 IF L=1.0 THEN 1540
1530 GOTO 250
1540 PRINT
1550 GOTO 1320
1560 PRINT "          SYSTEME D'EQUATIONS A 2 INCONNUES"
1570 PRINT :PRINT " AX + BY = E "
1580 PRINT " CX + DY = F"
1590 PRINT :INPUT "ENTREZ LES PARAMETRES A,B,C,D,E,F";A,B,C,D,E,F
1600 M=(A*D)-(B*C)
1610 IF M=0.0 THEN 1670
1620 X=((E*D)-(B*F))/M
1630 Y=((A*F)-(E*C))/M
1640 PRINT :PRINT "SOLUTION ", " X = ";X, " Y = ";Y
1650 PRINT "*****"
1660 GOTO 1690
1670 PRINT :PRINT "IL N'EXISTE PAS DE SOLUTION, OU PAS DE SOLUTION UNIQUE"
1680 PRINT "*****"
1690 PRINT :INPUT "POUR CONTINUER TAPEZ SUR 1, SINON SUR 0 ";L
1700 IF L=1.0 THEN 1720
1710 GOTO 250
1720 PRINT
1730 GOTO 1590
1740 PRINT "          NOMBRE PREMIER "
1750 REM this program tests if a number is prime
1760 REM it continues to cycle until zero is entered
1770 PRINT :PRINT "ENTREZ LE NOMBRE A TESTER, ZERO POUR STOPPER"
1780 INPUT N
1790 IF N=0.0 THEN 1950
1800 IF N<4.0 THEN 1910
1810 I=0.0
1820 T=2.0
1830 J=INT(N/T)
1840 K=J*T
1850 IF N=K THEN 1930
1860 I=I+1.0
1870 L=T*T
1880 IF L>N THEN 1910
1890 T=(I*2.0)+1.0
1900 GOTO 1830
1910 PRINT :PRINT " ";N;" EST PREMIER"
1920 GOTO 1770
1930 PRINT :PRINT N;" N'EST PAS PREMIER ";T;" EST LE PLUS PETIT FACTEUR"
1940 GOTO 1770

```