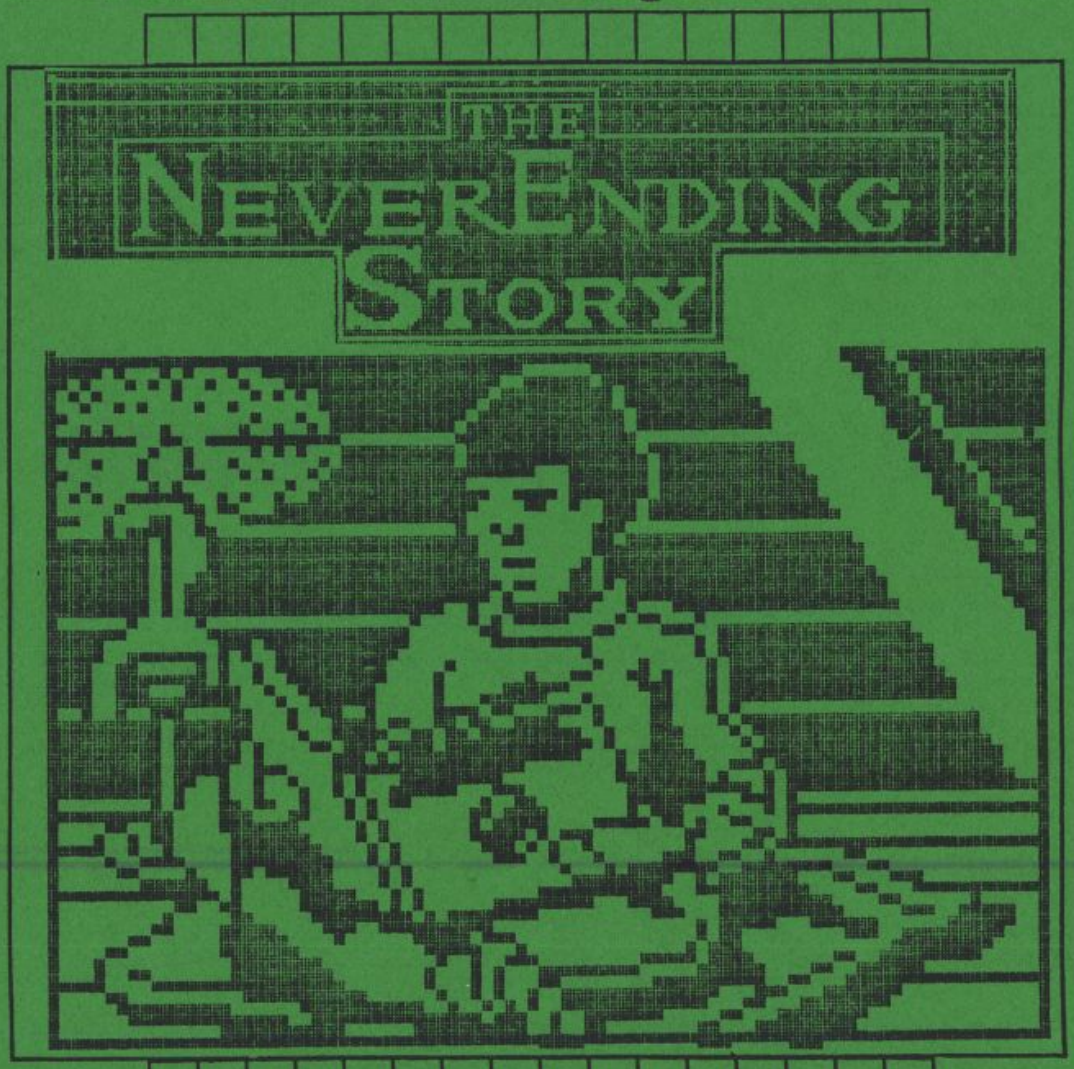
BULLETIN

SINCLAIR GEBRUIKERSGROEP GRONINGEN/ASSEN



6e jaargang nummer 2 october '88

COLOFON

VOORZITTER:
Jan Dirk Burggraaf
Kluivingskampenweg 30
9761 BP Eelde
☎ 05907-1697

SEKRETARIS:
Martin den Hollander
Numero Dertien 8
9644 TV Veendam
☎ 05978-45474

PENNINGMEESTER:
S.E. Kroon
Oosterhoutstraat 96
9401 NK Assen
☎ 05920-15912
Giro 5212298 t.n.v.
rekening SGG

VICE VOORZITTER/
PENNINGMEESTER:
J. van Alteren
De Grouw 6
9351 LP Leek
☎ 05945-15678

VERHUUR:
C. van Krimpen
Koldakker 34
9407 BM Assen
☎ 05920-70093

ALGEMEEN:
Roelof Koning
Selwerderstraat 26
9717 GK Groningen
☎ 050-124298

REDAKTIE:
Mevr. F. Elstrodt
Kam. Onnesstraat 172
9727 HS Groningen
☎ 050-263930

Rudy Biesma
Betuwe 18
9405 JJ Assen
☎ 05920-50643

Het SGG-bulletin is een uitgave van de Sinclair Gebruikersgroep Groningen. Het bulletin verschijnt 10 keer per jaar.

Artikelen, listings of andere inzendingen zijn voor verantwoording van de inzender.

De sluitingsdatum voor kopij wordt in elk bulletin vermeld.

Overname van artikelen, illustraties en andere publikaties uitsluitend toegestaan met toestemming van de redaktie.

Het lidmaatschap van onze gebruikersgroep bedraagt f 17,50 per kalenderjaar voor personen tot en met 17 jaar voor oudere personen is dit f 25,00 per kalenderjaar. Bij deze prijs is het abonnement op het bulletin inbegrepen.

U kunt lid worden van de SGG door U op te geven bij de penningmeester.

SLUITINGSDATUM KOPIJ 2 NOVEMBER

VAN DE REDAKTIE



HALLO ALLEMAAL

Mooie voorplaat van de Never Ending Story, Ramon verteld U over deze adventure meer op pagina 9.
U mag ten alle tijde bij U artikel, beschrijving etc. zelf Uw screen$ bedenken, dan wordt ons bulletin nog mooier.
Dit bulletin volgt zeer snel op de vorige en voor de redaktie was het even hard werken, maar we dachten er weer in geslaagd te zijn een leuk nummer te hebben gebracht.
Even een tip: wilt U de sluitingdatum die op pagina twee staat in de gaten houden als het copy voor die maand betreft.
Op onze vorige gebruikersavond ging ons cassette bandje van de cassette service gretig van computer tot computer. Dit schijnt onze leden wel aan te spreken, mooie programma's zonder zere vingers.
Binnenkort krijgen we een mooi fietscross programma wat geschreven is met Beta Basic 4.0 van Arie Lautenbach.
Onze Vice voorzitter wil beginnen met een Machine-Code-cursus en onze Sekretaris begint nu over Arrays, dus dat ziet er bitskleurig uit voor U lezers.

Deze maal zijn voor ons achter de computer gaan zitten:

- Onze voorzitter die gelukkig weer wat opgeknapt is wil zijn tuin in Masterfile, en dat kan ook nog.
- Roelof Koning met een prachtige IBM karakterset in TWdrie.
- Arie Lautenbach die je laat zien welke dag het wordt, hopelijk met veel vrije dagen.
- Ramon v. Ateren verteld over Adventures en geeft allerlei tips om te ontsnappen.
- Martin den Hollander rijdt met auto dwars door de woestijn,. en komt via een pyramide weer terug. Gelukkig.
- Han v. Abbe bespreekt de methodiek en routines die de gevraagde getallen genereren.
- Frans Postma laat alles even flitsen in deze donkere dagen.
- En Frans onderschept foutmeldingen in Basic.

In dit bulletin wordt Uw aandacht gevraagd voor onze hard en software markt, dus duik eens in de kasten en sleep uw overbodige spectrum spulletjes mee naar de gebruikersavond.
Vaak doet U er een ander een hoop genoegen mee.
Veel leesplezier en

TOT ZIENS OP DE CLUB.



RUDY & FLORA.

GEBRUIKERSAVOND GRONINGEN

SINCLAIR HARD- EN SOFTWARE MARKT

Onze eerst volgende gebruikersavond is 13 October, wij houden dan onze hard en softwaremarkt.
Het adres is: School "de Weijert"
van Schendelstr 1
Groningen.

GEBRUIKERSAVOND ASSEN

In het buurthuis 't Markehuus is de hele week Open huis. Op woensdag de 12de October houden wij van 20.00 tot 22.30 uur onze gebruikersavond.

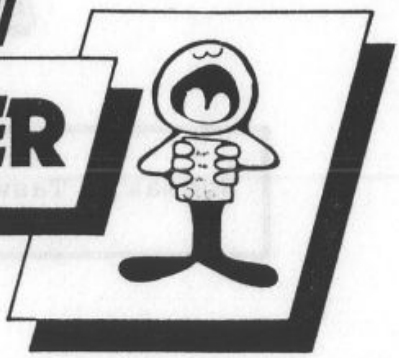


Tot ziens in:
Het Markehuus
Scharmbarg
Assen (Peelo = Wijk 7)

KOPIJ

Redaktie leden hebben altijd artikelen listings en leuke screen$ nodig.
Ook vinden wij het fijn, aardig commentaar te lezen of te horen, wilt U iets veranderd zien in ons bulletin hou ons dan op de hoogte, en probeer ook eens een bijdrage te leveren.
U weet nu zo langzamerhand wel hoe de kopij in te leveren.
En hoe wordt het met de rubriek KIJK OP SPELLETJES?.

VAN DE VOORZITTER



Een beetje opgeknapt en dan nu maar weer achter het toetsenbord om jullie weer van wat leesvoer te voorzien. Welkom bij het begin van alweer het zesde jaar dat onze Sinclair club bestaat. De avonden zijn alweer van dien aard dat de duisternis er voor zorgt dat er vroeg gestart kan worden met het beoefenen van de hobby.
De tuin, als die aanwezig is, kan alleen maar in het weekend verzorgd worden, maar natuurlijk zijn er genoeg mensen die al een nieuw plan maken voor het voorjaar. Misschien doen ze dit wel met behulp van de computer. Uiteindelijk zijn er prachtige programma's die hun hele hebben en houden kunnen opslaan.
Wat een mogelijkheden zijn er al niet. Precies bijhouden welke bloemen er in het voorjaar bloeien. Hoe hoog de planten van een bepaalde soort kunnen worden. Wanneer ze geplant of gezaaid moeten worden. Is dit niet allemaal prachtig voor een database (Masterfile bijvoorbeeld)?
Zo zijn er nog wel een aantal toepassingsmogelijkheden te bedenken voor het apparaat wat ons vaak tijdenlang in de ban houdt.
Het is natuurlijk wel leuk als jullie hierop willen reageren met een stukje bij de redactie in te leveren over de hulp die de computer je oplevert bij b.v. een andere hobby.
We zien met belangstelling jullie reacties tegemoet.

J.D. Burggraaf.



Nogmaals Tasword III en de IBM karakterset

Het is bekend dat alle goede dingen langzaam komen, en dus kwam ik er nu pas toe om het programma van Bert Koning (inderdaad familie) uit het bulletin van maart 1988 nuttig te gaan gebruiken.
Ik heb de data in TASTABLE gepoked, nog een paar wijzigingen aangebracht omdat ik ook graag de 'é' en de 'è' wilde gebruiken, en de nieuwe TASTABLE gesaved.
Toen kon ik beginnen met het aangeven van de kodes voor uitvoer naar de printer. Dat viel nog even tegen, zevenenzeventig nummers opzoeken in m'n printerhandboek, terwijl al die graphics zo verdraaid veel op elkaar lijken! Maar nu draait het spul, en oh wat gaat dat mooi!
Ter illustratie heb ik even iets in elkaar gezet waarvan je hiernaast twee afdrukken kunt zien, met, ter vergelijking, een screendump van het oorsponkelijke TASWORD-scherm.
Omdat bijna alle printers wel een IBM karakterset bezitten en de code's standaard zijn, geef ik hieronder een lijstje waar uit af te lezen valt welke code bij welk karakter hoort, en welke toets ervoor ingedrukt moet worden. Ook de bijbehorende DATA-regel (uit Bert's programma) is vermeld.

TOETS	CHAR	CODE	REGno.s	TOETS	CHAR	CODE	REGno.s
space		219	601/901	F	ƒ	159	639/939
!	┌	218	602/902	G	∞	236	640/940
"	┬	194	603/903	H	▌	221	641/941
#	┐	191	604/904	I	▐	222	642/942
$	├	195	605/905	J	⌠	244	643/943
%	┼	197	606/906	K	▄	223	644/944
&	└	192	607/907	L	▄	220	645/945
'	┴	193	608/908	M	╬	206	646/946
(	┘	217	609/909	N	**	252	647/947
)	┤	180	610/910	O	Ω	234	648/948
*	═	205	611/911	P	π	227	649/949
+	±	241	612/912	Q	Φ	232	650/950
,	‖	186	613/913	R	δ	235	651/951
-	—	196	614/914	S	σ	229	652/952
.	│	179	615/915	T	τ	231	653/953
/	÷	246	616/916	U	µ	230	654/954
0	°	248	617/917	V	ø	237	655/955
1	½	171	618/918	W	√	251	656/956
2	²	253	619/919	X	╡	181	657/957
3	ε	238	620/920	Y	╨	202	658/958
4	¼	172	621/921	Z	**	146	659/959
5	╒	201	622/922	^	♦	4	660/960
6	╤	209	623/923	_	♠	5	661/961
7	╕	187	624/924	`	╞	204	662/962

8	╚	200	625/925
9	╝	188	626/926
:	≡	240	627/927
;	▒	178	628/928
<	≤	243	629/929
=	≈	247	630/930
>	≥	242	631/931
?	**	177	632/932
@	┴	207	633/933
A	α	224	634/934
B	β	225	635/935
C	Γ	226	636/936
D	∩	239	637/937
E	Σ	228	638/938

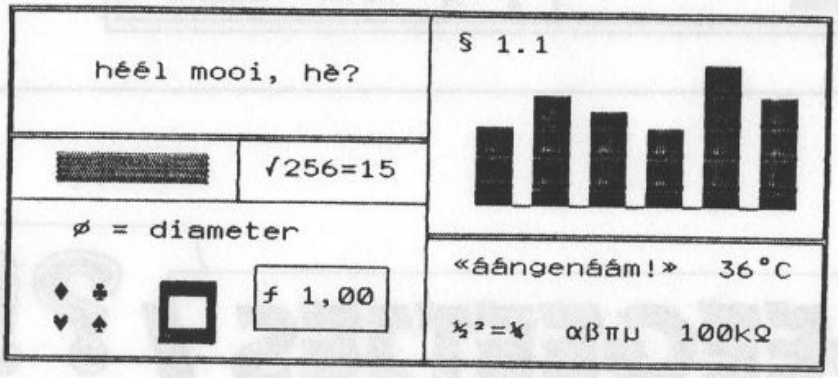
a	á	160	663/963
b	╡	185	664/964
c	ï	139	665/965
d	╞	199	666/966
e	ë	137	667/967
f	▒	182	668/968
g	╥	203	669/969
h	┸	176	670/970
i	«	174	671/971
j	»	175	672/972
k	♥	3	673/973
l	♠	6	674/974
m	╞	198	675/975

** = niet afgebeeld hier

Misschien kan de redaktie, voor wie deze uitbreiding toch onmisbaar moet zijn, het programma in een komplete vorm nog eens bij de casette-tape-service te doen?

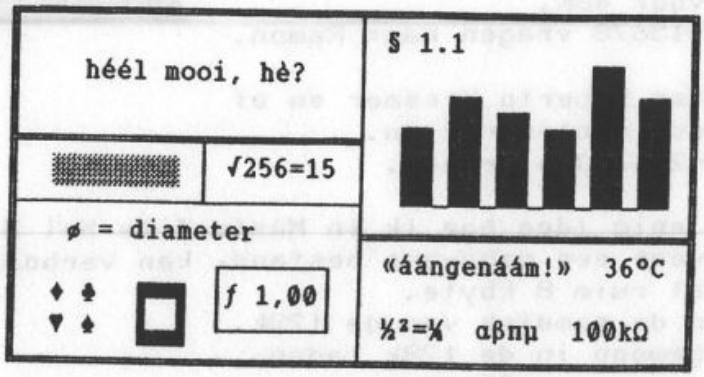
Roelof Koning.

P.S. Toch niet iedere printer blijkt over een IBM-karakterset te beschikken, o.a. de trouwe Smith-Corona laat het afweten, evenals (naar het schijnt) de Brother M1009.



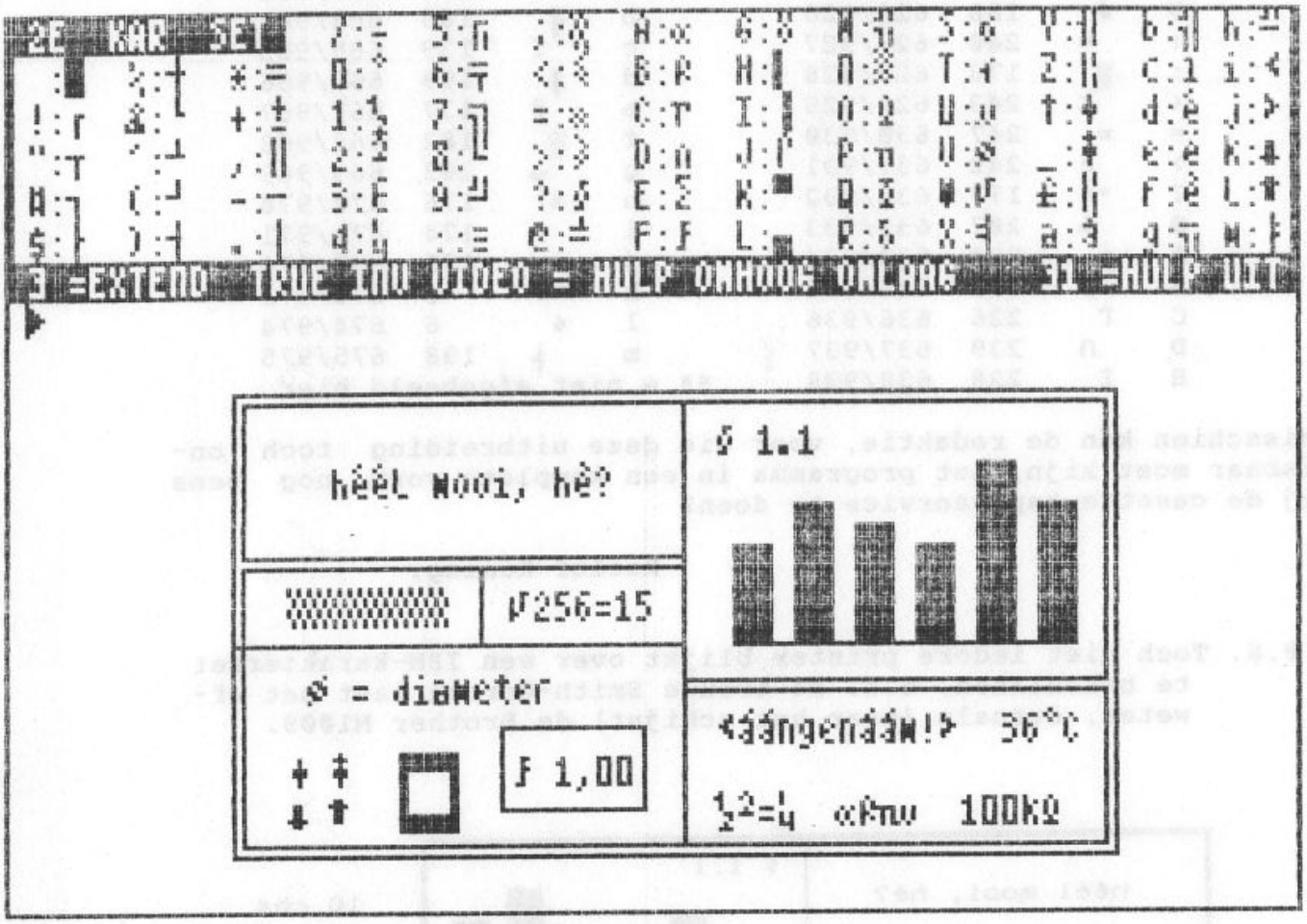
10 cps

draft



12 cps

NLQ



ESGEEGEETJES ?! !?

GEVRAAGD: Specdrum voor 48K.
Tel:05945-15678 vragen naar Ramon.

OPROEP: Wie weet waar Roberto Kreemer en of
Sina de Goede gebleven zijn.
Tel: 05985-2247 Jan Arends.

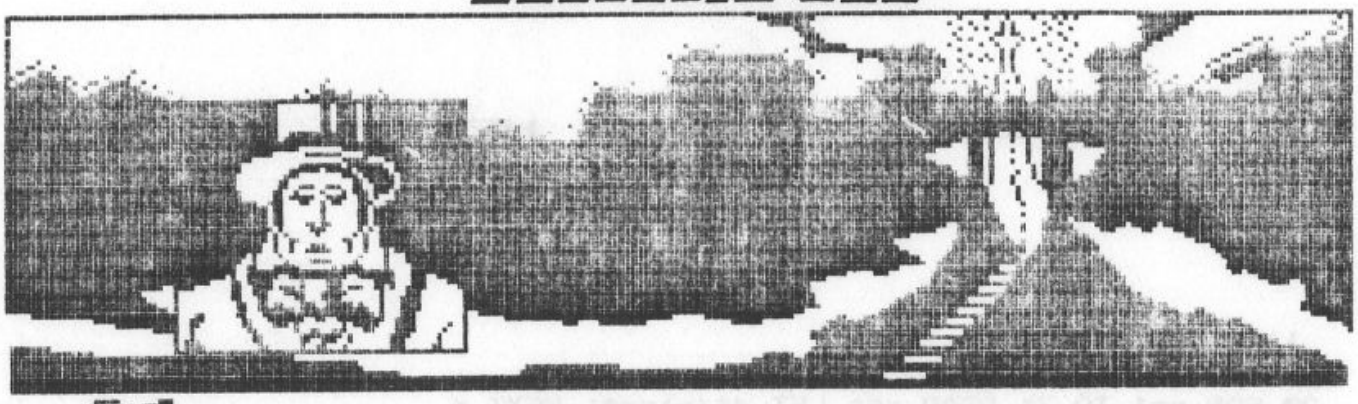
GEVRAAGD: Wie heeft enig idee hoe ik in Masterfile het 32 kbyte wat beschikbaar is voor een gegevens bestand, kan verhogen.
Het programma beslaat ruim 8 kbyte.
Ik dacht hierbij aan de ramdisk van de 128k.
Het program kun je gewoon in de 128k laden.
Reacties en tips graag naar Flora.

WELKE DAG IS HET ?

```
  1>REM  Dagbepaling
 10 DIM d$(7,9): DIM m$(12,2)
 20 FOR n=1 TO 7: READ r$: LET d$(n)=r$: NEXT n
 30 FOR n=1 TO 12: READ r$: LET m$(n)=r$: NEXT n
 40 LET dag=0: INPUT "jaar<nnnn>";jaar
 50 IF jaar/4=INT (jaar/4) THEN  LET m$(2)="29"
 60 IF jaar/4<>INT (jaar/4) THEN LET m$(2)="28"
 70 INPUT "maand  ";MAAND: IF MAAND<1 OR MAAND>12 THEN GO T
O 70
 80 INPUT "datum  ";DATUM: IF DATUM>VAL (M$(MAAND)) THEN GO
 TO 80
 90 IF MAAND>=2 THEN FOR N=1 TO MAAND-1: LET DAG=DAG+VAL M$
(N): NEXT N
 100 LET A=INT ((JAAR-1980)/4): IF (JAAR-1980)/4-A>0 THEN LE
T A=A+1: REM A IS AA
     NTAL SCHRIKKELJAREN
 110 LET DAG=DAG+DATUM+2+A*366+(JAAR-1980-A-1)*365
 120 LET dag=dag-7*INT (DAG/7)
 130 IF DAG=0 THEN LET DAG=7
 140 PRINT D$(DAG)
 150 PRINT #1;AT 0,0;"druk 'n toets": IF INKEY$="" THEN GO T
O 150
 160 CLS : GO TO 40
 170 DATA "maandag","dinsdag","woensdag","donderdag","vrijda
g","zaterdag","zonda
     g"
 180 DATA "31","28","31","30","31","30","31","31","30","31",
"30","31"
```

Een handig programma'tje van Arie Lautenbach.





You dre deep in the Forest. A small track runs southwest dnd to the east the Forest appedrs to thin out.
Rockbiter is here.

ADVENTURES

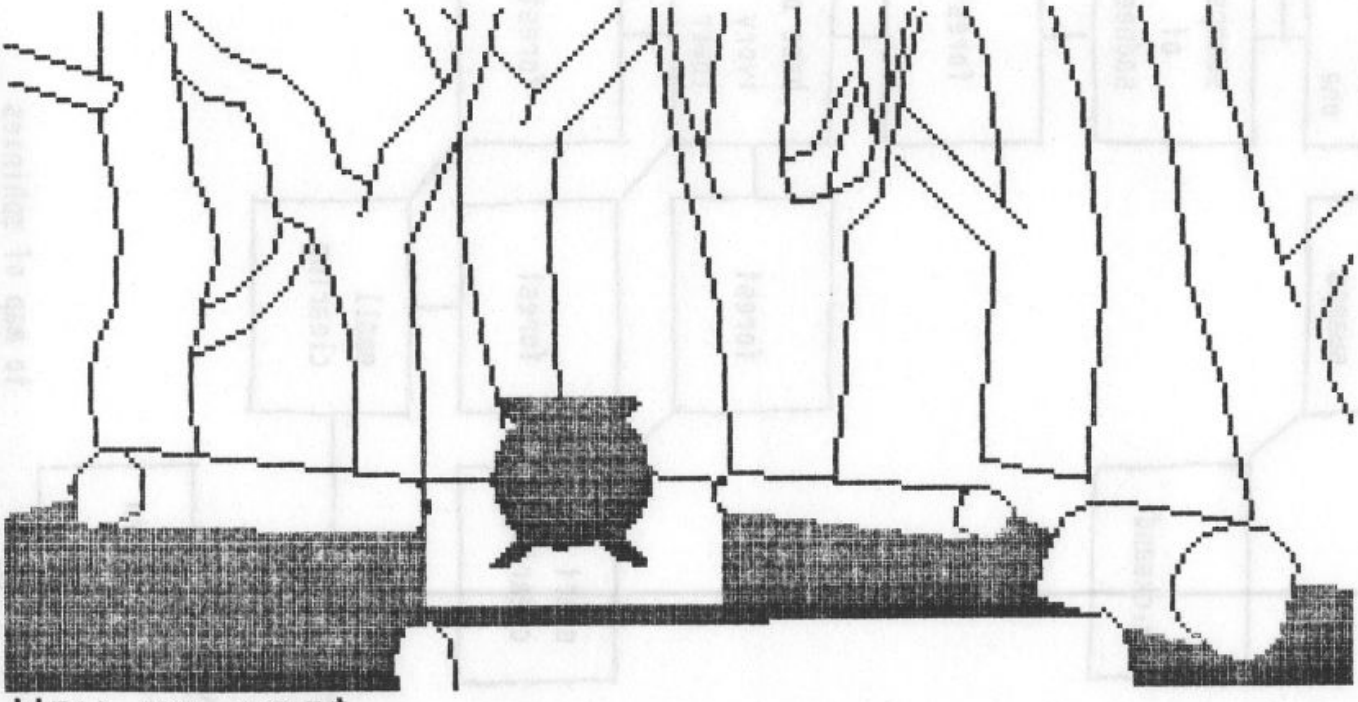
Het grootste probleem met adventures is dat ze in het engels geschreven zijn. De puzzels zijn zo al moeilijk genoeg en als je ze ook nog in het engels moet oplossen !
De logische oplossing zou zijn om ze naar het nederlands te vertalen. Dat is dan ook al eens geprobeerd.
In het begin van MCN is er een vertaling verkocht van Fantasia Diamond. Gek genoeg is de adventure in het nederlands moeilijker dan de engelse versie hoewel er behalve de taal geen verschil is. Een andere adventure die veel verkocht is,is 'The NeverEnding Story' (NES) van Ocean. Deze adventure kreeg men gratis bij aankoop van een Spectrum 128K
De adventure bestaat uit 3 delen en gaat over de held Atreyu.
Het verhaal is afgeleidt van de film en de film is weer gemaakt op basis van een boek. Het doel van Atreyu is om Fantasia weer in volle glorie te herstellen nadat het bijna volledig is opgegeten door het Niets.
Het spel begint met de vraag of je een geSAVEd spel wil inladen.
Het scherm is in twee delen gedeeld : boven een tekening van waar je nu bent en met kleine tekeningen van de voorwerpen die je in je bezit hebt.
Op de onderste helft kan je je opdrachten intypen en komt de beschrijving te staan. De karakterset is geredefineerd wat meestal tot het gevolg heeft dat de tekst niet meer te lezen is maar in dit geval valt dat wel mee.
De woordenschat van de computer is niet al te groot n.l. maar 39 woorden. Er ontbreekt een EXAMINE wat erg jammer is.
Een ander nadeel is dat de toetsgevoeligheid te hoog is afgesteld met het gevolg dat je vaak ggeet braanchh krijgt i.p.v get branch. De beschrijvingen van je omgeving zijn vrij goed en de problemen zijn niet al te moeilijk dus is deze adventure redelijk geschikt voor een beginner.
Het is verstandig om een kaart te maken .
Dit is vooral nuttig als je met een brandende tak rondloopt om wat lastig struikgewas te gaan verbranden en je kan de weg niet meer terug vinden.
De moeite van het tekenen van een kaart voor het eerste deel kunt u zich als het goed is besparen omdat die hierbij afgedrukt zou moeten worden.

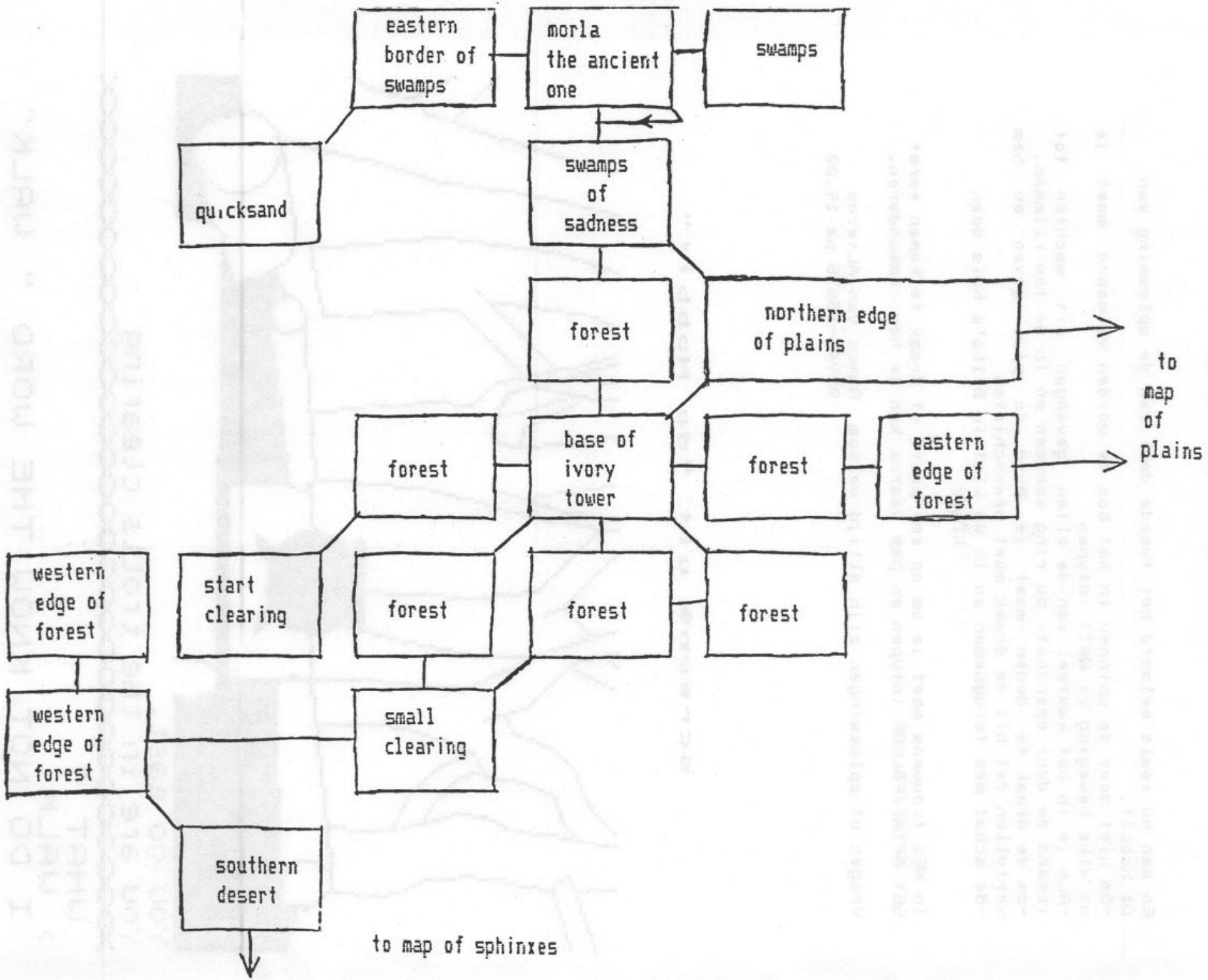
En dan nu zoals beloofd het tweede deel van de oplossing van De Hobbit.
-Om niet door de spinnen in het bos te worden vermoord moet je na elke beweging 2x WAIT intypen
-ALs je in het kasteel van de elfen gevangen zit wachten tot iemand de deur opendoet, de ring aandoen en in de ton klimmen.
-om de draak te doden moet je Bard de pijl geven en hem vertellen dat hij de draak moet neerschieten
-de schat mee terugnemen en in de kist in Bilbo's huis doen klaar.

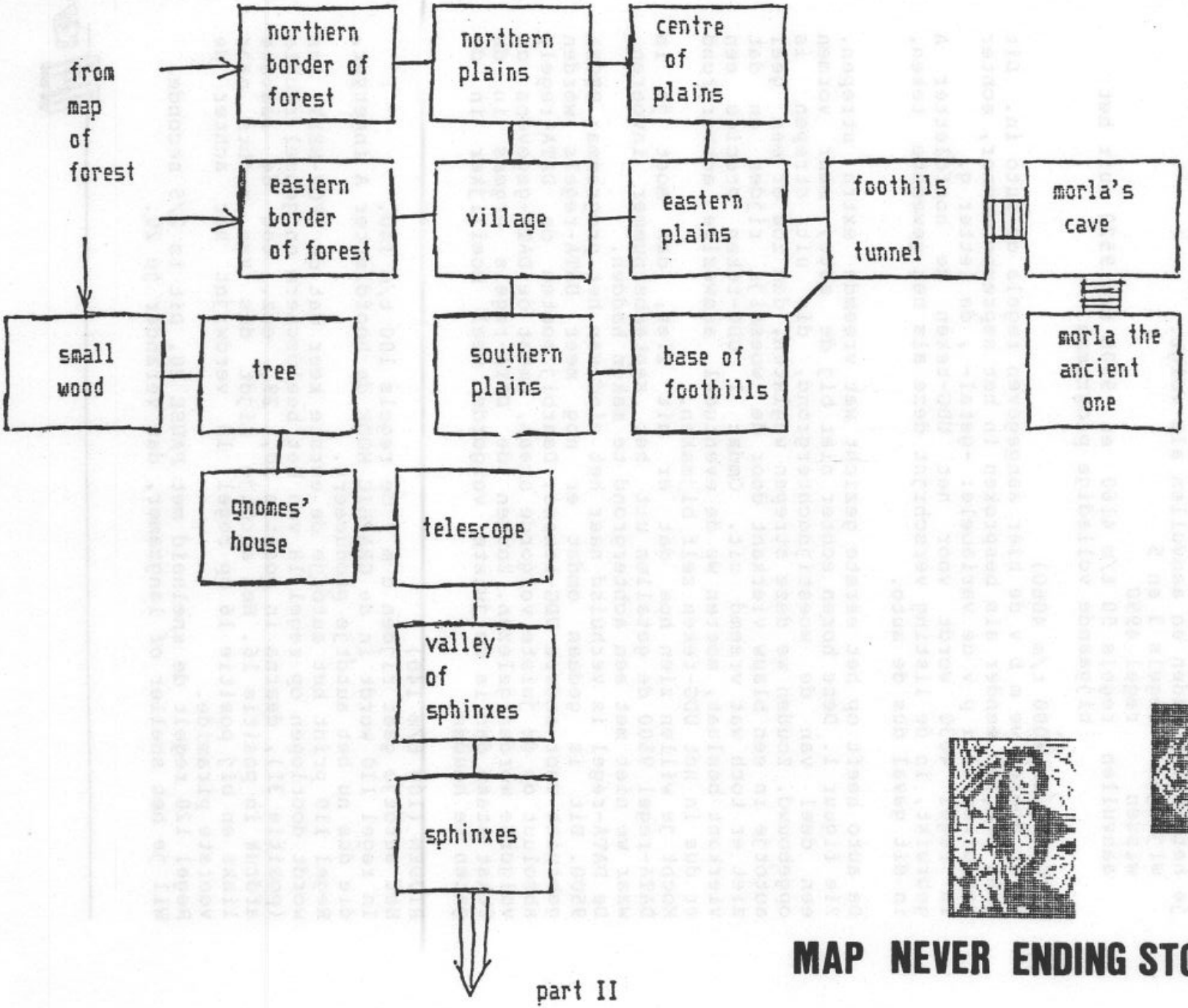
In NES trouwens moet je om op een paard of draak te komen eerst GET ARTAX/FALKOR intypen en pas daarna kan je hem commanderen.

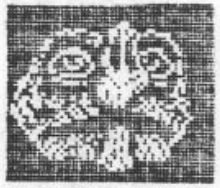
Vragen of oplossingen zijn altijd welkom Ramon van ALteren
05945-15678 na 19.00

screen$ uit "de Hobbit"




eastern border of swamps
morla the ancient one
swamps
quicksand
swamps of sadness
forest
northern edge of plains
to map of plains
forest
base of ivory tower
forest
eastern edge of forest
western edge of forest
start clearing
forest
forest
forest
western edge of forest
small clearing
southern desert
to map of sphinxes





MAP NEVER ENDING STORY

BASISPROGRAMMA'S VOOR BEGINNERS 5

AUTO TUSSEN DE PIRAMIDES VAN GIZEH (vervolg)

Deze keer zullen we het d m v een UDG ontworpen autótje van september tussen de piramides van Gizeh - mei '88 - laten doorrijden.

AANVULLING OP HET MEIPROGRAMMA

Als je het programma uit het meinummer '88 van het SGG-bulletin indertijd op cassette of cartridge hebt gezet, kun je het gewoon laden en aanvullen als volgt:

wijzigen	regels 3 en 5
wissen	regel 4990
aanvullen	regels 50 t/m 4160 en 9500 t/m 9520 uit het bijgaande volledige programma

AUTO (50 en 4000 t/m 4060)

Eerst voeren we m b v de hier aangegeven regels de auto in. Dit gaat precies eender als besproken in het septembernummer, echter gebruiken we i p v de variabele: -getal- , de letter q.
In regel 4030 wordt voor het UDG-teken de hoofdletter A gebruikt, in de listing verschijnt deze als het gewenste teken, in dit geval dus de auto.

De auto heeft op het eerste gezicht wat vreemde extra strepen. Zie figuur 1. Deze horen echter niet bij de auto, maar vormen een deel van de woestijnachtergrond, die uit strepen is opgebouwd. Zouden we deze strepen weglaten, dan zou er een geel autótje in een blauw vierkant door de woestijn rijden en dat ziet er toch wat vreemd uit. Omdat 1 UDG-teken precies een vierkant beslaat, moeten we de eventueel aanwezige achtergrond er dus in het UDG-teken zelf bijmaken.
Mocht je willen zien hoe dat er uit ziet, dan moet je in DATA-regel 9500 de getallen uit het septembernummer invoeren, waar we niet met een achtergrond te maken hadden.
De DATA-regel is verhuisd naar het eind van het programma: REGEL 9500. Dit is gedaan omdat er nog meer DATA-regels worden gebruikt voor andere UDG-tekens. Daarbij moeten de DATA-regels absoluut op de juiste volgorde staan, omdat de DATA-gegevens op volgorde worden gelezen. Zouden de DATA-regels ergens in de tekst staan dan is de juiste volgorde veel moeilijker in de gaten te houden.

RIJDEN (100 t/m 140)

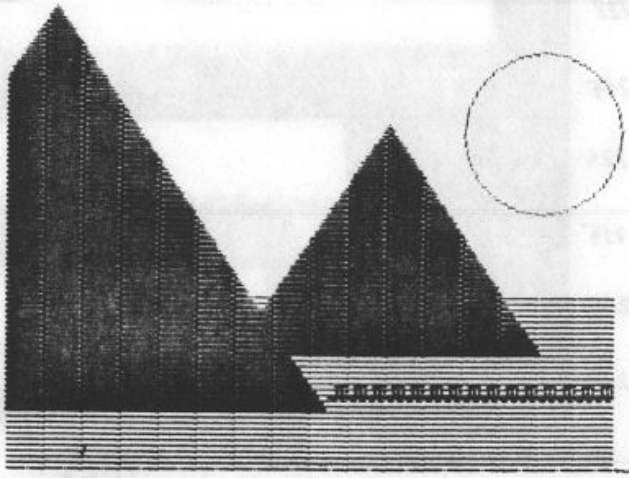
Het autótje gaat rijden d m v de regels 100 t/m 140.
In regel 110 wordt in de GRAPHIC MODE de hoofdletter A ingetypt, die dus nu het autótje produceert.
Regel 110 print het autótje de eerste keer dat de FOR-NEXT lus wordt doorlopen op regel 18 van het beeldscherm en geheel rechts (positie 31), daarna in positie 30, 29, enz. tot de laatste afdruk in positie 16. Het autótje rijdt dus van rechts naar links en bij positie 16 op regel 18 verdwijnt het achter de voorste piramide.
Regel 120 regelt de snelheid met PAUSE 20. Dit is 2/5 seconde. Wil je het sneller of langzamer, dan verander je 20.

Tot nu toe hebben we alleen een UDG-teken afgedrukt en als we er verder niets aan doen, komt er een file autótjes in de woestijn te staan. Zie figuur 2.
Door regel 130 wordt er een stukje woestijn afgedrukt op de plaats waar 2/5 seconde tevoren het autótje werd afgedrukt.
Dit stukje woestijn is dus een apart UDG-teken, dat natuurlijk eerst gemaakt moet worden. Dat gebeurt via regel 60 in de regels 4100 t/m 4160 en DATA-regel 9510. Zie figuur 3.
We gebruiken hier voor de FOR-NEXT variabele de letter o, voor de READ-instructie de variabele r en voor het UDG-teken de hoofdletter B.

Je zult misschien denken, hoe weet die READ-instructie van regel 4120 nu, dat hij DATA-regel 9510 moet hebben?
Dat weet hij natuurlijk niet, maar een READ-instructie pakt gewoon het allereerste DATA-gegeven, dat nog niet gelezen is.
Door READ q in regel 4020 zijn de 8 getallen van DATA-regel 9500 gelezen en READ r in regel 4120 begint bij het eerstvolgende DATA-gegeven en dat is het eerste getal in DATA-regel 9510.
Je zou ook alle DATA-gegevens in 1 DATA-regel achter elkaar kunnen zetten, dat maakt niets uit. (PROBEREN!)
De werking van regels 4100 t/m 4160 is verder precies hetzelfde als van regels 4000 t/m 4060.

HERSTEL PIRAMIDE
Tenslotte is nog een UDG-teken nodig omdat zowel de UDG-auto als de UDG-woestijn beide als vierkant worden afgedrukt. Het laatste vierkant (regel 18, positie 16) staat half in de piramide en dus moeten hier de woestijnlijnen schuin naar linksboven verlopend eindigen.
Je ziet weer eenzelfde programmaatje in de regels 200 t/m 260 en met de DATA-regel 9520. De gebruikte variabelen zijn s, p en C.
Het is in het algemeen verstandig steeds nieuwe variabelen te gebruiken, omdat je nooit weet, wat voor eindwaarde deze van een vorige bewerking hebben overgehouden.

De getallen in DATA-regel 9520 zijn door uitproberen op het scherm gevonden.
Tenslotte kun je het autótje achter de voorste piramide laten draaien en weer achter de achterste piramide naar rechts laten rijden, waar het rechtsboven in de woestijn dan weer te voorschijn komt en rechts het beeld uitrijdt.
De uitwerking hiervan wordt graag aan de lezer overgelaten, met enig plussen en minnen mag dat geen problemen meer opleveren.
Kom je er toch niet uit, dan wil ik je op de gebruikersavond graag helpen.

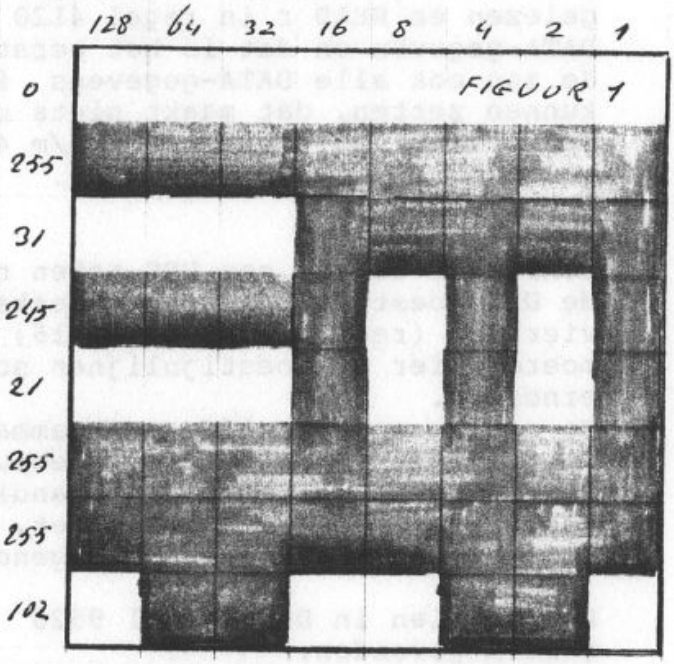


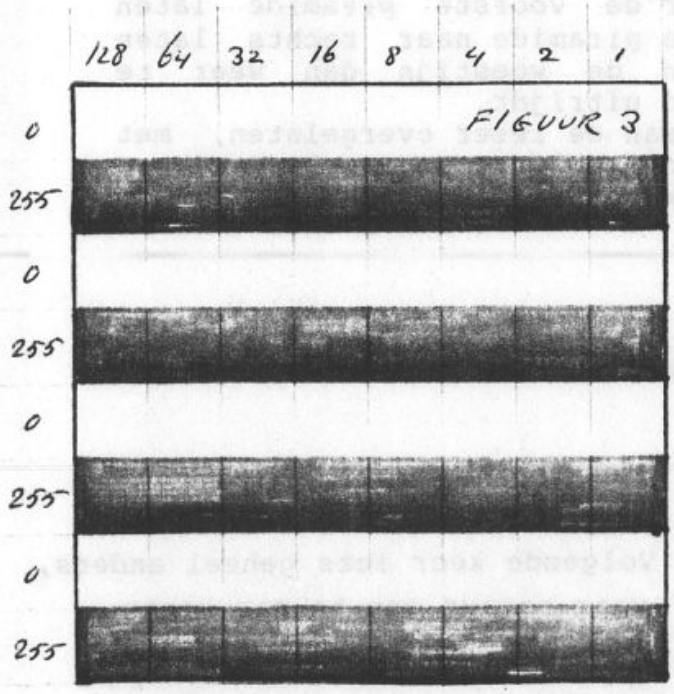
figuur 2

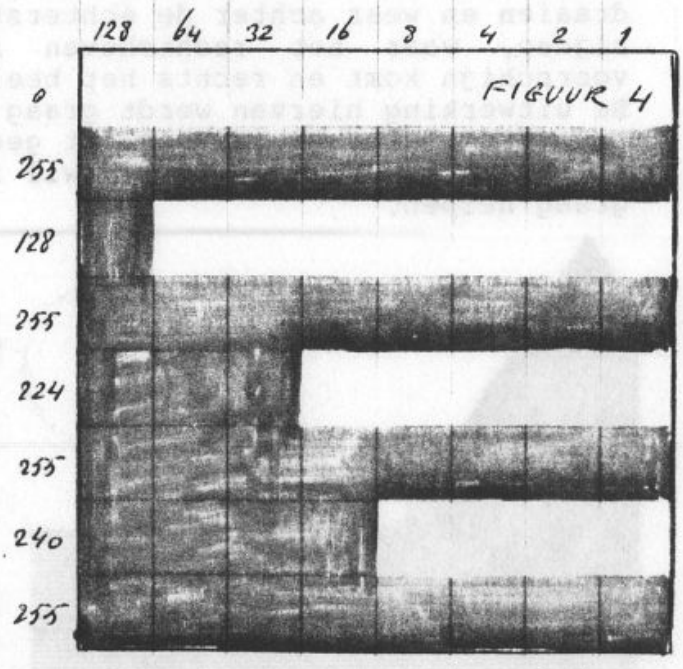
Volgende keer iets geheel anders, waar arrays aan te pas komen.

Martin den Hollander

```
   3 REM Titel "auto/gizeh"
   5 REM PIRAMIDEN VAN GIZEH MET
 AUTO DOOR WOESTIJN
  10 GO SUB 5000: REM WOESTIJN
  20 GO SUB 6000: REM KLEINE PIR
AMIDE
  30 GO SUB 7000: REM GROTE PIRA
MIDE
  40 GO SUB 8000: REM MAAN
  50 GO SUB 4000: REM AUTO
  60 GO SUB 4100: REM WOESTIJN A
CHTER AUTO
 100 FOR k=31 TO 16 STEP -1
 110 PRINT AT 18,k;"■"
 120 PAUSE 20
 130 PRINT AT 18,k;"≡"
 140 NEXT k
 200 REM HERSTEL PIRAMIDE
 210 FOR p=0 TO 7
 220 READ s
 230 POKE USR "E"+p,s
 240 NEXT p
 260 PRINT AT 18,16;"E"
3990 GO TO 9999
4000 REM AUTO
4010 FOR n=0 TO 7
4020 READ q
4030 POKE USR "■"+n,q
4040 NEXT n
4060 RETURN
4100 REM WOESTIJN ACHTER AUTO
4110 FOR o=0 TO 7
4120 READ r
4130 POKE USR "≡"+o,r
4140 NEXT o
4160 RETURN
5000 REM  WOESTIJN
5010 BORDER 0: PAPER 1: INK 6: C
LS
5020 FOR Y=0 TO 60 STEP 2
5030 PLOT 0,Y
5040 DRAW 255,0
5050 NEXT Y
5100 RETURN
6000 REM KLEINE PIRAMIDE
6010 FOR Z=-65 TO 65
6020 PLOT 160,120
6030 DRAW Z,-80
6040 NEXT Z
6100 RETURN
7000 REM GROTE PIRAMIDE
7010 FOR T=-24 TO -140 STEP -1
7020 PLOT 20,160
7030 DRAW -20,T
7040 NEXT T
7110 FOR X=-20 TO 115
7120 PLOT 20,160
7130 DRAW X,-140
7140 NEXT X
7200 RETURN
8000 REM MAAN
8010 CIRCLE 210,130,30
8020 RETURN
9500 DATA 0,255,31,245,21,255,25
5,102
9510 DATA 0,255,0,255,0,255,0,25
5
9520 DATA 0,255,128,255,224,255,
240,255
9999 BORDER 7: PAPER 7: INK 0
```







SNEL&PRECIES REKENEN MET DE ZX 81 (17)

Dit keer weer een Probleem van de Maand en wel uit de HCC Nieuwsbrief nr. 85 van september 1986. Mijn oplossing bevat naast varianten op al eerder besproken subroutines, elementen die ook in andere cijferproblemen bruikbaar zijn. De opgave luidde:
Zoek getallen van 9 cijfers, die
-de cijfers 1 t/m 9 precies een keer bevatten;
-geschreven kunnen worden als het produkt van 3 getallen van ieder 3 cijfers, waarbij deze getallen samen ook precies de cijfers 1 t/m 9 1 keer bevatten.
Ik heb het programma daarom
"ALL 9"
genoemd.

De oplossing verscheen in HCC Nieuwsbrief nr. 88 van december 1986. Het gepubliceerde program was in BBC-Basic voor de Acorn Electron computer. Dat PG deed er 8h 43min 9s over. Het hier behandelde vindt de oplossingen in 11min 13s, bijna 47-maal zo snel. In dit artikel bespreek ik de methodiek en de routines die de gevraagde getallen genereren. De volgende keer die antwoorden zoals ze worden gevonden en een Sorteerprogramma om de resultaten in de juiste volgorde te plaatsen.

METHODIEK

De opgave kan als volgt worden "vertaald":
-uit de cijfers 1 t/m 9 kunnen 9*8*7 = 504 getallen (G) worden gevormd met 3 van die cijfers;
-gevraagd wordt kombinaties van drie getallen G1, G2 en G3 te vinden, zodanig dat zo'n "set" alle cijfers 1 t/m 9 bevat en ook het resultaat R = G1*G2*G3 alle cijfers 1 t/m 9 enkelvoudig bevat.

Het aantal permutaties van 9 cijfers is 9! = 362880. Aangezien iedere G zowel op de eerste als op de tweede als op de derde plaats kan staan, komt iedere kombinatie 3! = 6 keer voor. Door te stellen G1>G2>G3 wordt het aantal zinvol te onderzoeken sets beperkt tot 60480. Daar de meeste computertijd (80% van het totaal) gaat zitten in het bijeen zoeken van die kombinaties, is nog een verdere beperking aangebracht. De kleinste waarde die R mag hebben is 123456789. Na enig proberen blijkt dat G1 nooit <612 en G2 nooit <412 kan zijn. Invoering van deze limieten in het programma vermindert het aantal te onderzoeken sets tot 44640.

PROGRAMMA

In figuur 1A staat het uiterst simpele BASIC-PG en in figuur 1B de HEXDUMP van de MC-RTN van regel 1. Figuur 2A toont een overzicht van de Stores Pointers en Tables, die zijn gebruikt en die gezamenlijk het RAM-gebied 4FF1-567F beslaan.

Figuur 2B geeft de MC-routine. MAKE TABLE 504 3-DIGIT-NUMBERS vult de G-TABLE met behulp van drie FOR/NEXT loops, funktioneel vertaald in MC. De registers A, B en C zijn voor respectievelijk de 1ste, 2de en 3de digit van G. D en E fungeren als opslagregisters. Het BASIC equivalent van dit stuk routine staat in figuur 2C. In BASIC kost het vullen van de G-TABLE ruim 16s, in MC 0.018s. Deze laatste routine werkt dus bijna 900-maal zo snel. Er worden twee bytes per G gebruikt. Zie figuur 3 die enkele gedeelten van de tabel (AD 5010-53FF) laat zien.

2B1

FIG. 1A BASIC-PG "ALL 9"

```
 1 REM-REGEL MET 336 BYTES
 2 RAND USR 16514
```

FIG. 1B HEXDUMP REGEL 1 FIG. 1A

```
4082--CDE7 0221 1050 1609--256
408A--7A5F 42B8 2816 4AB9--314
4092--280F 78B9 280A 7323--230
409A--A717 1717 1781 7723--21E
40A2--7B0D 20EB 0520 E43D--2D9
40AA--20DF 2110 5422 FB4F--2F0
40B2--210F 5023 EB21 30AE--28D
40BA--19EB D8CD B241 2B4E--415
40C2--2346 ED43 0A50 22F6--30B
40CA--4FED 53F1 4F23 EB21--3FE
40D2--50AD 1930 052A F64F--2BA
40DA--18D9 EBCD B241 3AF2--4C8
40E2--4FA2 20E9 3AF1 4FA3--417
40EA--20E3 ED4B F14F EB09--46F
40F2--22F3 4FEB 2B4E 2346--331
40FA--ED43 0C50 22F8 4F23--318
4102--7EA7 2005 2AF8 4F18--2D3
410A--C4CD B241 3AF4 4FA2--4A3
4112--20ED 3AF3 4FA3 20E7--433
411A--2B4E 2346 ED43 0E50--270
4122--E5CD 5141 7EA7 2824--3B5
412A--CDB2 410E 03CD B941--398
4132--0D20 FA21 01FF ED52--387
413A--E120 C4E5 2100 50ED--408
4142--5BFB 4F01 1000 EDB0--353
414A--ED53 FB4F E118 B021--454
4152--FF4F 1100 5001 0A00--1BA
415A--EDB0 2A0A 5022 0350--296
4162--623A 0D50 CD88 413A--2C9
416A--0C50 CD93 412E 091E--252
4172--0406 037E 1236 002D--100
417A--1D10 F83A 0F50 CD88--313
4182--413A 0E50 180B 4FCD--218
418A--9141 791F 1F1F 1FE6--2AD
4192--0F47 2E09 1E04 1A8E--157
419A--2777 2D1D F298 4110--2C3
41A2--F12E 0106 04AF ED6F--335
41AA--2D86 772C 2C10 F6C9--351
41B2--1100 007E CDC6 412C--28F
41BA--7ECD C341 7E1F 1F1F--32A
41C2--1FE6 0FC8 47AF 3717--320
41CA--10FD 3001 1C82 57C9--2FC
```

```
16514-CDE7 0221 1050 1609- 598
16522-7A5F 42B8 2816 4AB9- 788
16530-280F 78B9 280A 7323- 560
16538-A717 1717 1781 7723- 542
16546-7B0D 20EB 0520 E43D- 729
16554-20DF 2110 5422 FB4F- 752
16562-210F 5023 EB21 30AE- 653
16570-19EB D8CD B241 284E-1045
16578-2346 ED43 0A50 22F8- 779
16586-4FED 53F1 4F23 EB21-1022
16594-50AD 1930 052A F64F- 698
16602-18D9 EBCD B241 3AF2-1224
16610-4FA2 20E9 3AF1 4FA3-1047
16618-20E3 ED4B F14F EB09-1135
16626-22F3 4FEB 2B4E 2346- 817
16634-ED43 0C50 22F8 4F23- 792
16642-7EA7 2005 2AF8 4F18- 723
16650-C4CD B241 3AF4 4FA2-1187
16658-20ED 3AF3 4FA3 20E7-1075
16666-2B4E 2346 ED43 0E50- 624
16674-E5CD 5141 7EA7 2824- 949
16682-CDB2 410E 03CD B941- 920
16690-0D20 FA21 01FF ED52- 903
16698-E120 C4E5 2100 50ED-1032
16706-5BFB 4F01 1000 EDB0- 851
16714-ED53 FB4F E118 B021-1108
16722-FF4F 1100 5001 0A00- 442
16730-EDB0 2A0A 5022 0350- 662
16738-623A 0D50 CD88 413A- 713
16746-0C50 CD93 412E 091E- 594
16754-0406 037E 1236 002D- 255
16762-1D10 F83A 0F50 CD88- 787
16770-413A 0E50 180B 4FCD- 536
16778-9141 791F 1F1F 1FE6- 685
16786-0F47 2E09 1E04 1A8E- 343
16794-2777 2D1D F296 4110- 707
16802-F12E 0106 04AF ED6F- 821
16810-2D86 772C 2C10 F6C9- 849
16818-1100 007E CDC6 412C- 655
16826-7ECD C341 7E1F 1F1F- 810
16834-1FE6 0FC8 47AF 3717- 800
16842-10FD 3001 1C82 57C9- 764
```

FIG. 2A STORES + TABLES "ALL 9"

```
STORE OF G1-BITS 9/8...1   4FF1/2
STORE OF G1- AND G2-BITS   4FF3/4

G1-AD(DRESS) IN G-TABLE    4FF6/7
G2-AD        IN G-TABLE    4FF8/9

PTR(POINTER) R-TABLE       4FFB/C

THE FOLLOWING TABLES/STORES ARE
WITH DEC(IMAL) DIG(ITS) HI...LO:

NUMBERS STORE              5000-500F
(DE): (00) 01/04     (HL): 05/09
G1: 0A/0B  G2: 0C/0D  G3: 0E/0F

G-TABLE OF 3-DIGIT-NRS 5010-53FF

R-TABLE WITH RESULTS   5410-567F
```

FIG. 2C BASIC EQUIVALENT VAN 1STE DEEL MC-ROUTINE

```
 10 LET D=20496
 20 FOR A=9 TO 1 STEP -1
 30 FOR B=9 TO 1 STEP -1
 40 IF B=A THEN GOTO 120
 50 FOR C=9 TO 1 STEP -1
 60 IF C=A OR C=B THEN GOTO 110
 70 POKE D,A
 80 LET D=D+1
 90 POKE D,16*B+C
100 LET D=D+1
110 NEXT C
120 NEXT B
130 NEXT A
```

FIG. 2B MACHINECODE-RTN "ALL 9"

2B1 MAKE TABLE 504 3-DIGIT-NUMBERS

```
          4082--CDE702   CALL  02E7       SET-FAST
          4085--211050   LD    HL,5010    G-TABLE
          4088--1609     LD    D,09       begin value LOOPS
          408A--7A       LD    A,D        A = 9....1
A-LOOP    408B--5F       LD    E,A        store A
          408C--42       LD    B,D        B = 9....1
B-LOOP    408D--B8       CP    B
          408E--2816     JR    Z,40A6     NEXT B if B=A
          4090--4A       LD    C,D        C = 9....1
C-LOOP    4091--B9       CP    C
          4092--280F     JR    Z,40A3     NEXT C if C=A
          4094--78       LD    A,B
          4095--B9       CP    C
          4096--280A     JR    Z,40A2     ;NXT C if C=B
          4098--73       LD    (HL),E     1, dig of G
          4099--23       INC   HL
          409A--A7       AND   A          → no carry
          409B--17       RLA
          409C--17       RLA
          409D--17       RLA
          409E--17       RLA              A = 2, dig of G
          409F--81       ADD   A,C        C = 3, dig of G
          40A0--77       LD    (HL),A
          40A1--23       INC   HL
NXT C     40A2--7B       LD    A,E        retrieve A
NEXT C    40A3--0D       DEC   C
          40A4--20EB     JR    NZ,4091    C-LOOP
NEXT B    40A6--05       DEC   B
          40A7--20E4     JR    NZ,408D    B-LOOP
NEXT A    40A9--3D       DEC   A
          40AA--20DF     JR    NZ,408B    A-LOOP
```

FIG. 3 GEDEELTEN VAN DE G-TABLE

```
5010--0987 0986 0985 0984--23A
5018--0983 0982 0981 0978--222
5020--0976 0975 0974 0973--1F6
5028--0972 0971 0968 0967--1D6
5030--0966 0964 0963 0962--1B2
5038--0961 0958 0957 0956--18A
................
........
51B8--0627 0625 0624 0623--0AB
51C0--0621 0619 0618 0617--081
51C8--0615 0614 0613 0612--066
51D0--0598 0597 0596 0594--26D
51D8--0593 0592 0591 0589--253
51E0--0587 0586 0584 0583--228
................
........
5298--0427 0426 0425 0423--0A5
52A0--0421 0419 0418 0417--079
52A8--0416 0415 0413 0412--060
52B0--0398 0397 0396 0395--256
52B8--0394 0392 0391 0389--24C
52C0--0387 0386 0385 0384--222
................
........
53D0--0154 0153 0152 0149--146
53D8--0148 0147 0146 0145--11E
53E0--0143 0142 0139 0138--0FA
53E8--0137 0136 0135 0134--0DA
53F0--0132 0129 0128 0127--0AE
53F8--0126 0125 0124 0123--096
```

2B2

In FIND SET OF 3 3-DIGIT-NUMBERS worden de kombinaties G1 G2 G3 bij elkaar gezocht uit de G-TABLE zodanig dat alle 9 cijfers verschillend zijn. Eerst wordt de PTR R-TABLE (4FFB/C) op 5410, het beginadres van die tabel, geSET. G1 wordt getest op de eerder genoemde limietwaarde 612 die zich op AD 51CE/F bevindt. Dit gebeurt door optelling van AE30 het komplement van het eerstvolgende AD 51D0 (G = 598). Als die waarde is bereikt wordt een carry gegenereerd en eindigt het PG via RET C, AD 40BC. G1 wordt geCONVERTeerd in BITs met de subroutine op AD 41B2/D1. G1 zelf wordt opgeslagen op 500A/B, het bijbehorende adres in de tabel op 4FF6/7 en het BIT-equivalent op 4FF1/2.

G2 wordt getest op de limietwaarde 412 (AD 52AE/F) door optelling van de hexwaarde AD50, komplement van het eerstvolgende AD 52B0 (G = 398). Als deze test een carry oplevert wordt HL weer op G1-AD gebracht en gaan we door met de volgende waarde van G1. Als G2 >= 412 is, worden na conversie de BITs vergeleken met die van G1. Indien alle 6 BITs verschillend zijn, worden deze opgeslagen op 4FF3/4, zo niet dan gaan we door met de volgende G2. Bij een goede set G1 G2 slaan we G2 op 500C/D op en het bijbehorende adres in de tabel op 4FF8/9.
De laagste waarde van G3 is 123, AD 53FF. Dit einde van de tabel is herkenbaar gemaakt met de test AD 4102/04 en de lege byte op AD 5400. Als dat punt bereikt is, wordt HL = G2-AD en wordt met de volgende waarde van G2 doorgegaan. Zo niet, dan wordt ook G3 in BITs omgezet, die nu worden vergeleken met de 6 BITs van G1 G2 op 4FF3/4. Indien er geen BITs zijn die coincideren, wordt G3 opgeslagen op 500E/F en gaan we G1*G2*G3 in het volgende deel van de RTN berekenen.

2B3

MULTIPLY + TEST RESULT: "ALL 9?" roept eerst de "echte" MULTIPLY-subroutine aan, waarover dadelijk meer. Deze RTN levert op 5005/9 het resultaat R = G1*G2*G3 op. De eerste -dus hoogste- digit van R op AD 5005 moet >0 zijn, anders bevat R slechts 8 digits en behoeven we dat getal niet verder te onderzoeken. Daarom de test op AD 4126/29. Als R>1E9 is worden de cijfers van R met de al genoemde CONVERT-subroutine in BITs omgezet, die "aangeleverd" worden in het DE-register. Die waarde vergeleken met HL = FF01 (in BITs LH=0000 0001 1111 1111) vertelt ons of R alle 9 cijfers bevat. Zo ja, dan gaan we naar het volgende routine-gedeelte, zo nee, dan wordt het zoeken voortgezet.

2B4

STORE RESULT AND FACTORS behoeft als RTN geen nadere toelichting. Per R worden op 10h bytes achtereenvolgens geSTOREd:
-het tussenresultaat (DE): G1*G2 na 3* SHIFT-4BITS-LEFT, op de plaatsen .0 t/m .4;
-het Resultaat (HL), plaatsen .5 t/m .9;
-G1, G2 en G3 op de plaatsen .A t/m .F.

De 1ste groep van de de volgende keer te behandelen R-TABLE ziet er dan ook als volgt uit:
5410--0553 7070 0001 2956--1B8
5418--7438 0987 0561 0234--1D8
dat is de oplossing die het eerst wordt gevonden.

2B5

MULTIPLY DEC 3 3-DIGIT-NUMBERS bestaat uit vier gedeelten. In het 1ste stuk worden de 10 bytes van de NUMBERS STORE geCLEARd. Het 3de en 4de stuk passen in vereenvoudigde vorm de vermenigvuldig-methodiek toe, die ik in deel (15) uitvoerig behandeld heb en ook in deel (16) heb toegepast:
-ADC decimaal (DE) B-maal naar (HL) met in (DE) het vermenigvuldigtal en B het aantal eenheden van de vermenigvuldiger;
-Shift (DE) 4 BITs Left en voer de MULT-BYTE-routine uit met voor B het aantal tientallen van die vermenigvuldiger;
-herhaal de procedure voor de honderdtallen.

Vereenvoudigingen die de reken-snelheid ten goede komen zijn mede het gevolg van de bewuste keuze van het (DE)-gebied eindigend met de LSB 04, waardoor DEC E/JP P (AD 419D/A0) voor de benodigde 5 rondes zorgt en tevens het werken met L en E in plaats van HL en DE mogelijk is. Verder is gebruik gemaakt van het feit dat geen van de drie digits van G een nul kan zijn.

Het 2de stuk van deze subroutine verzorgt de "logistiek". Eerst G1 naar (DE) en vermenigvuldigen met G2-LO (2 digits) en G2-HI (1 digit). Het tussenresultaat gaat dan van (HL) naar (DE), (HL) wordt geCLEARd en daarna vermenigvuldigen we met respectievelijk G3-LO en G3-HI.

2B2 FIND SET OF 3 3-DIGIT-NUMBERS

```
SET-R-PTR 40AC--211054    LD   HL,5410     R-TABLE
          40AF--22FB4F    LD   (4FFB),HL  PTR R-TABLE
          40B2--210F50    LD   HL,500F    G-TABLE -1
G1        40B5--23        INC  HL
          40B6--EB        EX   DE,HL      DE = AD in TABLE
          40B7--2130AE    LD   HL,AE30    see text
          40BA--19        ADD  HL,DE
          40BB--EB        EX   DE,HL      HL = AD in TABLE
          40BC--D8        RET  C          to BASIC if G1 < 612
          40BD--CDB241    CALL 41B2       CONV-3-NIB
          40C0--2B        DEC  HL
          40C1--4E        LD   C,(HL)
          40C2--23        INC  HL
          40C3--46        LD   B,(HL)
          40C4--ED430A50  LD   (500A),BC  G1-STORE
          40C8--22F64F    LD   (4FF6),HL  G1-AD
          40CB--ED53F14F  LD   (4FF1),DE  G1-BITS
G2        40CF--23        INC  HL
          40D0--EB        EX   DE,HL      DE = AD in TABLE
          40D1--2150AD    LD   HL,AD50    see text
          40D4--19        ADD  HL,DE
          40D5--3005      JR   NC,40DC    CONT if G2>398
          40D7--2AF64F    LD   HL,(4FF6)  G1-AD
          40DA--18D9      JR   40B5       G1
CONT      40DC--EB        EX   DE,HL      HL = AD in TABLE
          40DD--CDB241    CALL 41B2       CONV-3-NIB
          40E0--3AF24F    LD   A,(4FF2)   BIT "9" - G1
          40E3--A2        AND  D          D = BIT "9" - G2
          40E4--20E9      JR   NZ,40CF    G2 if coïncide
          40E6--3AF14F    LD   A,(4FF1)   BITs "8...1" - G1
          40E9--A3        AND  E          E = BITs "8...1" - G2
          40EA--20E3      JR   NZ,40CF    G2 if coïncide
NO_COINCIDENCE 40EC--ED4BF14F LD BC,(4FF1) G1-BITS
          40F0--EB        EX   DE,HL      HL = BITs G2
          40F1--09        ADD  HL,BC
          40F2--22F34F    LD   (4FF3),HL  G1 & G2 - BITS
          40F5--EB        EX   DE,HL      HL = AD in TABLE
          40F6--2B        DEC  HL
          40F7--4E        LD   C,(HL)
          40F8--23        INC  HL
          40F9--46        LD   B,(HL)
          40FA--ED430C50  LD   (500C),BC  G2-STORE
          40FE--22F84F    LD   (4FF8),HL  G2-AD
G3        4101--23        INC  HL
          4102--7E        LD   A,(HL)
          4103--A7        AND  A          see text
          4104--2005      JR   NZ,410B    NOT-END
          4106--2AF84F    LD   HL,(4FF8)  G2-AD
          4109--18C4      JR   40CF       G2
NOT_END   410B--CDB241    CALL 41B2       CONV-3-NLB
          410E--3AF44F    LD   A,(4FF4)   BIT "9"-G1&G2
          4111--A2        AND  D          D = BIT "9" - G3
          4112--20ED      JR   NZ,4101    G3 if coïncide
          4114--3AF34F    LD   A,(4FF3)   BITs "8..1"-G1&G2
          4117--A3        AND  E          E=BITs "8...1"-G3
          4118--20E7      JR   NZ,4101    G3 if coïncide
          411A--2B        DEC  HL
          411B--4E        LD   C,(HL)
          411C--23        INC  HL
          411D--46        LD   B,(HL)
          411E--ED430E50  LD   (500E),BC  G3-STORE
```

2B3 MULTIPLY + TEST RESULT: "ALL 9?"

```
          4122--E5        PUSH HL         stack G3-AD
MULT      4123--CD5141    CALL 4151       CLEAR/MULTIPLY
          4126--7E        LD   A,(HL)     HL = 5005
          4127--A7        AND  A          (HL) = 00 ?
          4128--2824      JR   Z,414E     TO-G2 if yes
ALL 9?    412A--CDB241    CALL 41B2       CONV-3-NIB
          412D--0E03      LD   C,03       3 rounds
NXT-2-NIB 412F--CDB941    CALL 41B9       CONV-2-NIB
          4132--0D        DEC  C
          4133--20FA      JR   NZ,412F    NXT-2-NIB
          4135--2101FF    LD   HL,FF01    LH=0000 0001 1111 1111
          4138--ED52      SBC  HL,DE      ED = R-BITS
          413A--E1        POP  HL         retrieve G3-AD
          413B--20C4      JR   NZ,4101    G3 if not ALL 9
```

2B4 STORE RESULT AND FACTORS

```
          413D--E5        PUSH HL         stack G3-AD
          413E--210050    LD   HL,5000    NUMBERS STORE
          4141--ED5BFB4F  LD   DE,(4FFB)  PTR-R-TABLE
          4145--011000    LD   BC,0010
          4148--EDB0      LDIR            see text
          414A--ED53FB4F  LD   (4FFB),DE  new value PTR
          414E--E1        POP  HL         retrieve G3-AD
TO-G3     414F--18B0      JR   4101       G3
```

2B5 MULTIPLY DEC 3 3-DIGIT-NUMBERS

```
CLEAR     4151--21FF4F    LD    HL,4FFF
          4154--110050    LD    DE,5000     (DE) + (HL)
          4157--010A00    LD    BC,000A       cleared
          415A--EDB0      LDIR

MULTIPLY  415C--2A0A50    LD    HL,(500A)   G1 digs
          415F--220350    LD    (5003),HL   to (DE)
          4162--62        LD    H,D         5 hex
          4163--3A0D50    LD    A,(500D)    G2-LO
          4166--CD8841    CALL  4188        MULT-BYTE
          4169--3A0C50    LD    A,(500C)    G2-HI
          416C--CD9341    CALL  4193        MULT-NIB
          416F--2E09      LD    L,09        HL = end (HL)
          4171--1E04      LD    E,04        DE = end (DE)
          4173--0603      LD    B,03        3 bytes
NX        4175--7E        LD    A,(HL)      from (HL)
          4176--12        LD    (DE),A      to (DE)
          4177--3600      LD    (HL),00     and clear (HL)
          4179--2D        DEC   L
          417A--1D        DEC   E
          417B--10F8      DJNZ  4175        NX
          417D--3A0F50    LD    A,(500F)    G3-LO
          4180--CD8841    CALL  4188        MULT-BYTE
          4183--3A0E50    LD    A,(500E)    G3-HI
          4186--180B      JR    4193        MULT-NIB

MULT-BYTE 4188--4F        LD    C,A         store A
          4189--CD9141    CALL  4191        RIGHT-NIB
          418C--79        LD    A,C         retrieve A
          418D--1F        RRA               left nibble
          418E--1F        RRA                 to
          418F--1F        RRA
          4190--1F        RRA
RIGHT-NIB 4191--E60F      AND   0F          Right nibble
MULT-NIB  4193--47        LD    B,A         A rounds
NXT       4194--2E09      LD    L,09        HL = end (HL)
          4196--1E04      LD    E,04        DE = end (DE)
NEXT-BYTE 4198--1A        LD    A,(DE)
          4199--8E        ADC   A,(HL)      see text
          419A--27        DAA
          419B--77        LD    (HL),A
          419C--2D        DEC   L
          419D--1D        DEC   E           5 rounds
          419E--F29841    JP    P,4198      NEXT-BYTE
          41A1--10F1      DJNZ  4194        NXT

SHIFT-4BITS-LEFT 41A3--2E01  LD   L,01      1 byte (DE)
          41A5--0604      LD    B,04        4 rounds
NXT-BYTE  41A7--AF        XOR   A
          41A8--ED6F      RLD
          41AA--2D        DEC   L
          41AB--86        ADD   A,(HL)
          41AC--77        LD    (HL),A
          41AD--2C        INC   L
          41AE--2C        INC   L
          41AF--10F6      DJNZ  41A7        NXT-BYTE
          41B1--C9        RET               HL = 5005
```

2B6 CONVERT DEC DIGITS INTO BITS 3 OR 2 NIBBLES

```
CONV-3-NIB 41B2--110000   LD    DE,0000     clear BIT-register
           41B5--7E       LD    A,(HL)
           41B6--CDC641   CALL  41C6        SET-BIT
CONV-2-NIB 41B9--2C       INC   L
           41BA--7E       LD    A,(HL)
           41BB--CDC341   CALL  41C3        RIGHT-DIG
           41BE--7E       LD    A,(HL)
           41BF--1F       RRA               Left
           41C0--1F       RRA                 digit
           41C1--1F       RRA                 to
           41C2--1F       RRA                 Right
RIGHT-DIG  41C3--E60F     AND   0F          if digit is 0
           41C5--C8       RET   Z
SET-BIT    41C6--47       LD    B,A         A rounds
           41C7--AF       XOR   A           clear A-register
           41C8--37       SCF               begin with carry
NXT-SHIFT  41C9--17       RLA
           41CA--10FD     DJNZ  41C9        NXT-SHIFT
           41CC--3001     JR    NC,41CF     NOT-"9"
           41CE--1C       INC   E           SET BIT "9"
NOT-"9"    41CF--82       ADD   A,D         SET BITS "0....1"
           41D0--57       LD    D,A
           41D1--C9       RET
```

286

CONVERT DEC DIGITS INTO BITS
3 OR 2 NIBBLES is qua essentie besproken in deel (2) van deze serie. De enige toevoeging is de "ingang" 3-NIBBLES, waarmee de test "ALL 9?" begint. Gevolgd door driemaal de 2-NIBBLES-RTN heben we dan de 9 nibbles van R geconverteerd.



R E S U L T A T E N

Na 673s blijken 39 goede kombinaties te zijn gevonden, waarvan de kleinste
127.386.945 = 945 * 827 * 163 is en de grootste
438.176.952 = 942 * 876 * 531.
Ten opzichte van de oplossing die ik van het vraagstuk in september 1986 had ingestuurd brengt het nu gepubliceerde PG "slechts" 33s verbetering in de rekentijd. De belangrijkste winst is verkregen met de test op AD 4126/29.

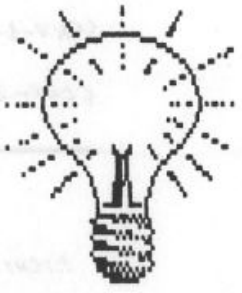
De RTN die de sets G1 G2 G3 vindt, welke voldoen aan de ALL 9?-test, gebruikt 9min 80% van de totale tijd! Indien iemand voor die routine tijdbesparende suggesties heeft, houd ik mij gaarne aanbevolen.

H A N V A N A B B E

FLASH

Dit programma werd ingestuurd door Frans Postma.
Typ het in en zie wat er gebeurt.

```
 10 CLS : REM FLASH ONN/OFF
 20 PRINT "FLASH SWITCH     F.POSTMA 1988"
 30 PRINT ''"INPUT ADRES:";: INPUT ADR: IF ADR+30>65367 OR ADR+3
0<23800 THEN PRINT FLASH 1;"ERROR": GO TO 30
 40 PRINT ADR: PRINT AT 21,10; FLASH 1;"POKING"
 50 DEF FN A(X$,B)=CODE X$(B)-48-7*(CODE X$(B)>57)
 60 DEF FN B(X$)=16*FN A(X$,1)+FN A(X$,2)
 70 REM DATA
 80 DATA "21","00","58","0E"
 90 DATA "02","06","C2","CB","7E"
100 DATA "20","04","CB","FE","28"
110 DATA "02","CB","BE","23","10"
120 DATA "F3","79","FE","00","C8"
130 DATA "0D","06","FF","18","EA"
140 DATA "00","21","00",F.P."
150 LET T=0: FOR F=ADR TO ADR+30
160 READ X$: LET X=FN B(X$): PRINT AT 10,9;(ADR+31)-F,: POKE F,X
: LET T=T+X: NEXT F
170 IF T<>3028 THEN PRINT "ERROR IN DATA !!!!": STOP
180 CLS : PRINT "POKED...."''"SAVE MET:CODE ";ADR;",30"''"RUNNEN
MET ~RANDOMIZE USR ";ADR;"~"
190 STOP
200 SAVE "FLASH" LINE 10
210 STOP
220 SAVE *"M";1;"FLASH" LINE 10
```



ON ERROR GOTO

```
  10 CLS : PRINT "BY ERROR GOTO LINEF.POSTMA '88"
  20 PRINT AT 2,0;"INPUT ADRES:";: INPUT ADR: IF ADR+73>65367 OR
ADR<23800 THEN PRINT FLASH 1;"ERROR": GO TO 20
  30 PRINT ADR: PRINT AT 4,0;"GOTO BY ERROR ?(0-9999):";: INPUT L
N: IF LN<0 OR LN>9999 THEN GO TO 30
  40 PRINT LN: PRINT AT 21,10; FLASH 1;"POKING"
  50 DEF FN A(X$,B)=CODE X$(B)-48-7*(CODE X$(B)>57)
  60 DEF FN B(X$)=16*FN A(X$,1)+FN A(X$,2)
  70 REM DATA
  80 DATA "CD","7E","00","3B","3B","E1"
  90 DATA "01","0F","00","09","EB","2A"
 100 DATA "3D","5C","73","23","72","C9"
 110 DATA "3B","3B","CD","8E","02","7B"
 120 DATA "FE","FF","20","F8","3A","3A"
 130 DATA "5C","FE","FF","28","21","FE"
 140 DATA "07","28","1D","FE","08","28"
 150 DATA "19","3C","32","81","5C","FD"
 160 DATA "36","00","FF","21","17","25"
 170 DATA "22","42","5C","AF","32","44"
 180 DATA "5C","FD","CB","01","FE","C3"
 190 DATA "7D","1B","33","33","C3","03"
 200 DATA "13","00",F.POSTMA"
 210 LET T=0: FOR F=ADR TO ADR+73
 220 READ X$: LET X=FN B(X$): PRINT AT 10,9;(ADR+73)-F,: POKE F,X
: LET T=T+X: NEXT F
 230 IF T<>7360 THEN CLS : PRINT "ERROR IN DATABLOCK !!!!!!!": ST
OP
 240 RANDOMIZE LN: POKE ADR+52,PEEK 23670: POKE ADR+53,PEEK 23671
: REM POKE LINE NUM.
 250 CLS : PRINT "POKED...."''"SAVE MET:CODE ";ADR;",73"''"RUNNEN
 MET ~RANDOMIZE USR ";ADR;"~"''"GEPOKED VOOR LINE:";LN
 251 PRINT FLASH 1;AT 13,0;"   CODE RUNNEN AAN BEGIN PROG",
 260 PRINT AT 21,7;"SAVE CODE....": INPUT "NAAM:";N$: IF LEN N$=0
 OR LEN N$>10 THEN GO TO 260
 270 SAVE N$CODE ADR,73: CLS
 280 STOP
 290 SAVE "ON ERROR" LINE 10
 300 SAVE *"M";1;"ON ERROR" LINE 10
```

Dit gebruikersprogramma onderschept foutmeldingen in Basic.
In geval van een fout wordt naar het opgegeven regelnummer gesprongen, zodat je de fout kunt herstellen.
Je kunt b.v. hiermee ook zorgen, dat ze in jouw programma niet kunnen BREAKen.
Het is ingezonden door Frans Postma.

DRUKWERK

PORT BETAALD
GRONINGEN

AFZ:
SGG
REDAKTIEADRES
Mevr.F.Elstrodt
Kam.Onnesstr 172
9727 HS Groningen

AAN:

Elstrodt, F
K. Onnestraat 172
9727 HS Groningen