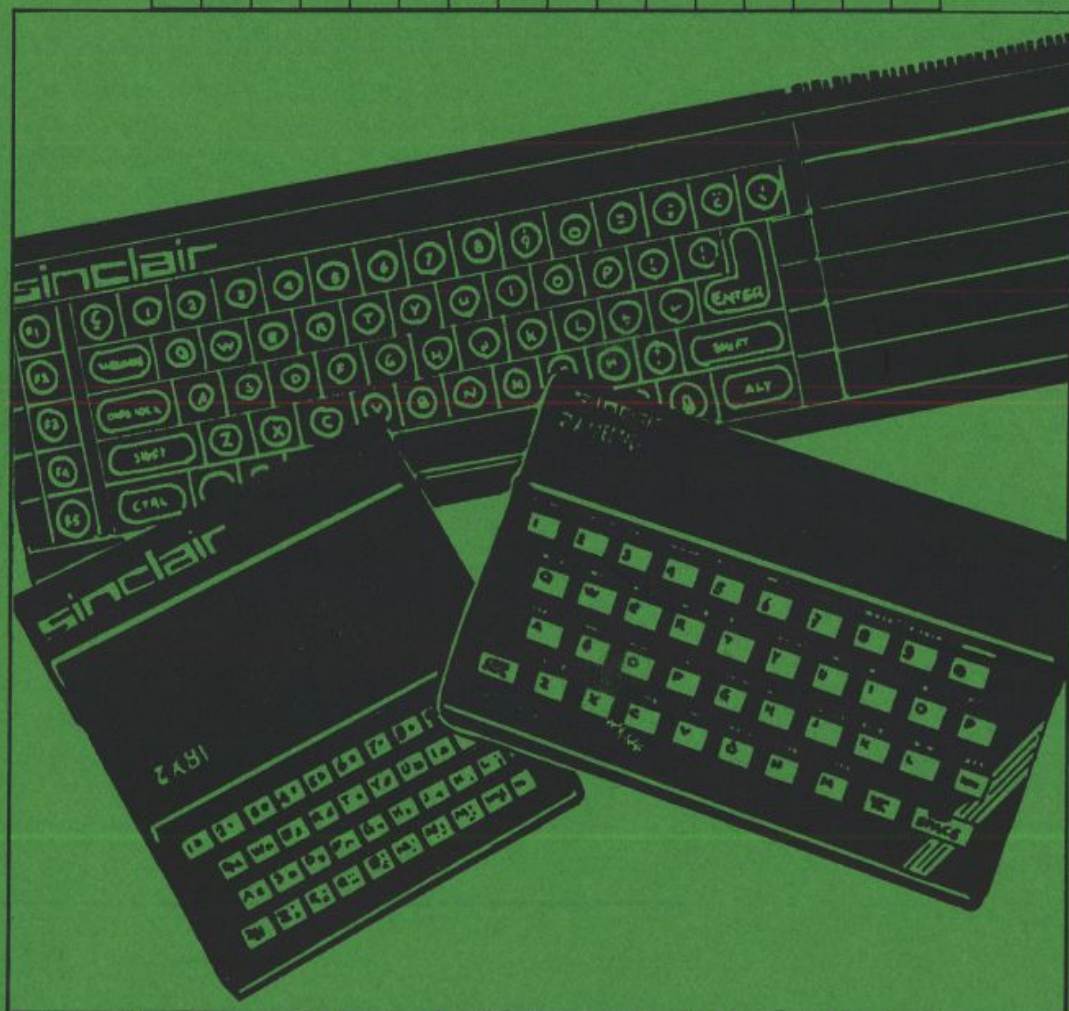


# BULLETIN

## SINCLAIR GEBRUIKERSGROEP GRONINGEN/ASSEN





# COLLOFON

## VOORZITTER:

Jan Dirk Burggraaf  
Kluivingskampenweg 30  
9761 BP Eelde  
tel. 05907-1697

## SEKRETARIS:

Martin den Hollander  
Numero Dertien 8  
9644 TV Veendam  
Tel. 05978-45474

## PENNINGMEESTER:

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

## VICE VOORZITTER/

## PENNINGMEESTER:

J. van Alteren  
De Grouw 6  
9351 LP Leek  
Tel. 05945-15678

## VERHUUR:

C. van Krimpen  
Koldakker 34  
9407 BM Assen  
Tel. 05920-70093

## REDAKTIE BULLETIN:

Jan R. Kloosterman  
Geert Valckeshof 28  
9351 RX Leek  
Tel. 05945-16328

Orm Heerkens  
Burg. van Waninglaan 33  
9351 LS Leek  
Tel. 05945-12669

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

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

De sluitingsdatum voor kopij wordt in elk bulletin vermeld.

Opgave voor lidmaatschap + abonnement op het bulletin à f. 17,50 per jaar bij de penningmeester (zie boven).

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

# VAN DE REDAKTIE



**H**elaas, het is afgelopen voor de redactie. Zoals al gemeld heeft Orm het de laatste tijd het erg druk met z'n PC met 2 drives, enz., en nu heeft Jan z'n Speccie ook in de wilgen gehangen en probeert de 1 Megabyte van zijn Atari vol te krijgen (en hij heeft al zo weinig vrije tijd!).

In het vorige nummer hebben we al aangekondigd dat dit zou gaan gebeuren, maar het kwijtraken van de Spectrum van Jan heeft het proces enigzins versneld. Het bulletin is nu ook grotendeels gemaakt op een geleende computer (bedankt, Marcel!) en dat kun je natuurlijk ook niet eeuwig blijven doen. Ook waren we in het vorige nummer bang dat door de matige (lees : geen) reacties het wel eens afgelopen zou kunnen zijn met jullie lijfblad. Gelukkig kunnen we nu meer optimistisch zijn, er hebben zich mensen aangemeld die het wel eens willen proberen. Daarover in het volgende nummer meer.

De ondergetekende redactie heeft er inmiddels (bijna) twee jaar opzitten en heeft dit met veel plezier gedaan (misschien krijgen we nog wel eens spijt). Het samenstellen van het Bulletin kende natuurlijk z'n ups en downs, bla, bla, bla ..... We willen gewoon iedereen langs deze sympathieke weg hartelijk bedanken voor de medewerking, ook de huisgenoten die we vaak overlast bezorgden door de meestal tot diep in de nacht durende luidruchtige redactievergaderingen. En natuurlijk ook voor de hulp bij het vouwen, nieten en verzenden.

Het Bulletin blijft dus bestaan en in dit nummer hebben we onder andere de volgende artikelen:

- Onze Spectrum (6)
- Geen Break
- Rekenen met de ZX 81 (10)
- Tips voor de spectrum

Veel leesplezier en tot ziens,

Jan en Orm.

## GRONINGEN

**D**e eerstvolgende avond in Groningen is op donderdag 26 november, en op deze avond demonstreerd Henk Boon zijn abusievelijk voor oktober vermelde stamboom-programma. Voor degenen die nu nog niet weten waar de gebruikersavonden zijn vermelden we nog maar een keer het adres:

School "de Wijert"  
van Schendelstraat 1  
Groningen

Voor de liefhebbers: de gebruikersavond in december is op de 10e, en dan hebben we weer de voor velen langverwachte Hard- en Softwaremarkt, waarover meer in dit nummer.



## ASSEN

**V**oor Assen begint de routine er wat in te komen. Als het zo doorgaat moet er misschien nog wel eens uitgekeken worden naar een groter zaaltje. De gebruikersavonden zijn elke 2e donderdag van de maand, en dat betekent dat de eerstvolgende avond op 12 november is. Deze happening vindt plaats in:

Het Markehuus

Walakker 11

Assen (Peelo = Wijk 7)

Allemaal komen en je apparatuur meenemen! In december is de datum de 10e, en ook daar wordt iedereen weer verwacht. Tot ziens!

## VAN DE VOORZITTER



**Z**o maar weer een paar gedachten die mij te binnen schieten. Het is maar goed dat er clubs zijn. Je kan tenminste je ideeën die je hebt uitspreken of laten zien op de clubavonden.

Ook blijken de problemen die je hebt met hard-of software, als sneeuw voor de zon te smelten als ze voorgelegd worden aan mede clubleden. De een wil het wel samen met je proberen op te lossen de ander heeft een pasklaar antwoord.

Het idee dat onze secretaris ontwikkeld heeft om zgn. instructeurs te vragen of ze bereid zijn de mede leden te helpen lijkt mij uitstekend.

Ik hoop dat er veel mensen zijn die zich hier voor aanmelden. Dit hoeft uiteraard niet alleen op de clubavond maar kan ook met een simpel telefoontje naar Martin den Hollander of naar een van de andere bestuursleden.

Als dit door jullie gelezen wordt is de 27e oktober inmiddels voorbij, waarbij ik hoop dat een ieder dit een zinnige avond vond en voor herhaling vatbaar is.

Wij als bestuur doen in ieder geval ons best en een positieve bijdrage van de leden is natuurlijk ook nooit weg.

Zijn er vragen op wat voor (Sinclair computer)gebied dan ook, schroom niet om ons op de avond aan te schieten of pak toch even de telefoon. Zelfs een berichtje in de PGSU databank kan natuurlijk ook.

J.D. Burggraaf

## INSTANT SCREEN

**E**r zijn de laatste tijd veel nieuwe leden bij onze gebruikersgroep gekomen die nog niet zo lang een Spectrum hebben of nu pas de mogelijkheden van de computer ontdekken. Om te ontdekken wat voor mogelijkheden de Spectrum heeft publiceren we ook regelmatig artikelen voor mensen die pas beginnen en natuurlijk ook voor de "oude rotten" die ook wel eens iets tegenkomen wat ze nog niet wisten.

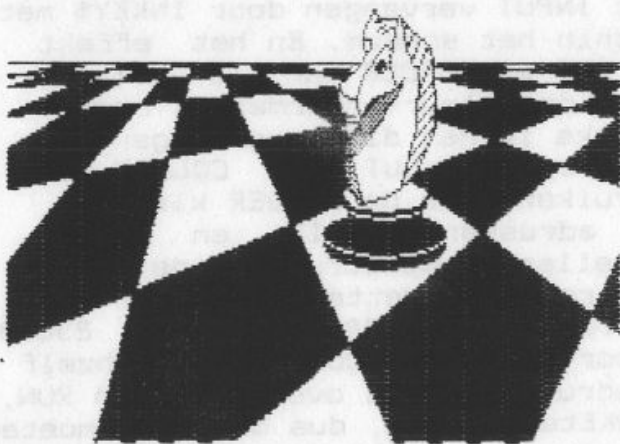
Deze keer gaan we het hebben over SCREEN\$. De 280 processor die in de Spectrum zit kan 64K geheugen tegelijk aan. Dat is totaal 65,536 bytes. Een byte wordt normaal gesproken gebruikt om een karakter op te slaan, en is de hoeveelheid geheugen die gemanipuleerd wordt door PEEK of POKE.

Je kunt het gemakkelijkst deze 65,536 bytes voorstellen als 256 bladzijden van elk 256 bytes. De eerste 64 bladzijden, genummerd van 0 tot 63, worden gebruikt voor de ROM van de Spectrum. De RAM zit op bladzijde 64 tot 255.

Een SCREEN gebruikt 27 bladzijden, en de bladzijden 64 - 90 zijn de pagina's die gebruikt worden voor LOAD "" SCREEN\$. De eerste 24 pagina's bevatten het monochrome raster van 256 x 192 puntjes en de laatste 3 bladzijden zijn voor de kleuren-informatie. Als je extra SCREENS in het geheugen wilt stoppen moet je geheugenruimte CLEARen om te voorkomen dat deze overschreven wordt door BASIC. Er is ruimte in het geheugen van de 48K Spectrum voor 5 SCREENS, en je houdt dan 7 K over voor een BASIC controle-programma'tje.

Het is verstandig deze 5 SCREENS te starten op de bladzijden 120, 147, 174, 201 en 228. Om ruimte te CLEARen voor 5 SCREENS type je CLEAR 120\*256-1. Als je maar 2 SCREENS wilt opslaan gebruik je CLEAR 201\*256-1. Deze twee moeten dan starten op blz. 201 en 228. Je kunt ook een SCREEN rectstreeks in het geheugen zetten met LOAD "" CODE pagina \*256,6912.

Zet pagina op 120, 147, 174, 201 of 228, welke plaats je ook wilt gebruiken. Bladzijde 64 zet het geladen SCREEN gelijk op je tv of monitor. De hieronder geplaatste korte routine past op bladzijde 255, vlak na het laatste SCREEN en net voor de UDG's.



```
10 CLEAR 120*256-1: REM (C) juni 1987 Simon N Goodwin
20 LET move=65280: LET from=56282: LET target=65285
30 RESTORE 50: FOR p=65280 TO 65291
40   READ x: POKE p,x: NEXT p
50 DATA 33,0,64,17,0,228,1,0,27,237,176,201
60 LIST
70 POKE from,64: POKE target,228: RANDOMIZE USR move
80 CLS: PAUSE 0
90 POKE from,228: POKE target,64: RANDOMIZE USR move
100 PAUSE 0: GO TO 80
```



De regels 10 t/m 50 van het voorbeeld-programma reserveren het geheugen en maken de machine code. Regel 60 zet iets op het scherm (je kunt ook LOAD ""SCREEN\$ gebruiken). Regel 70 SAVEd de display op bladzijde 228, terwijl regel 80 het scherm weer schoonmaakt en wacht op een toetsindruk. Regel 90 brengt het weer terug van bladzijde 228 naar het display-geheugen (bladzijde 64) en regel 100 wacht weer op een toetsdruk om het allemaal weer over te doen. Het bladzijdenummer van de te verplaatsen bladzijde kan gemakkelijk worden veranderd door een nieuwe waarde in 65288 te POKEn. Meer over dit onderwerp vindt je in hoofdstuk 24 van het originele (oranje) handboek van de Spectrum. Probeer het allemaal maar eens!

(Bron: Crash)

## GEEN BREAK

**B**reakie, breakie. Hoe kun je je zelfgemaakte programma beschermen tegen inbraak? Heel eenvoudig! POKE 23659 met 0 en je voorkomt dat de Spectrum het onderste gedeelte van het beeldscherm (het INPUT gebied) kan gebruiken. Foutmeldingen worden hier neergezet, dus als je BREAK indrukt kan er geen foutmelding worden neergezet en springt je Spectrum op tilt. Dit truukje heeft nadelen voor drie statements: INPUT, CLS en BORDER. Deze gebruiken ook het onderste gedeelte van het scherm en kunnen dus niet gebruikt worden. Geen probleem! Je kunt INPUT vervangen door INKEY\$ met een printopdracht ergens bovenin het scherm. En het effect van CLS en BORDER kun je krijgen door de kleur-informatie rechtstreeks in het displaygeheugen te POKEn en door OUT 254, COLOUR te gebruiken voor de BORDER kleur.

De adressen 23613 en 23614 vertellen de Spectrum hoe de zaken weer recht te zetten in geval van een fout (zoals BREAK). POKE 23613,4 : POKE 23614,61 zorgt ervoor dat de computer zichzelf reset als BREAK wordt ingedrukt. Helaas overschrijven RUN, CLEAR, GO SUB en RETURN de gePOKEte waarden, dus de POKEs moeten door het hele programma na een van deze commando's herhaald worden.



## KOPIJ

**Z**eker nu is het belangrijk om kopij in te sturen! De nieuwe redactie zal alle steun nodig hebben die ze kunnen krijgen. Stuur voor het volgende nummer artikelen, tekeningen, spelbesprekingen of wat dan ook op naar het redactie-adres. Maar wel graag voor 26 november. Inleveren op de eerstvolgende gebruikersavond mag ook. Kopij kan worden ingeleverd op cassette, cartridge of 3 1/2 " (Opus) disk. Bij voorbaat hartelijk bedankt!



## ONZE SPECTRUM (6)

**Z**oals ik in deel 5 al schreef, is het mogelijk om een andere charset te gebruiken in de Spectrum. Ook is het mogelijk om een charset van het ene programma te gebruiken in een ander programma zoals van Tassword en Basicode 2 en 3. Zelf heb ik de charset van Basicode 2 versie Ament in Tassword 2 gezet, hetgeen de leesbaarheid duidelijk bevordert. Maar dit is niet zomaar een kwestie van overzetten. Tassword maakt n.l. gebruik van een 64 koloms routine, waarvoor de karakters, om toch op het scherm te kunnen, nogal verkleind moeten worden. Basicode gebruikt een 42 koloms routine, hetgeen minder aanpassing vraagt. Als we dan ook kijken naar het ontwerp van beide sets zien we duidelijke verschillen.

We nemen als voorbeeld de letter A. Zoeken we hiervan de waarden in de charset,  $A(65) - \text{spatie}(32) = 33 \times 8 = 264$  bytes hoger dan het begin + 8, dan vinden we in Tassword 0, 2, 5, 5, 7, 5, 5, 0 : in Basicode 3 : 16, 40, 68, 68, 124, 68, 68, 0 : in Basicode Ament 0, 24, 36, 36, 60, 36, 36, 0. Laten we dit even in een matrix zetten.

7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
0	16	0
2	40	24
5	68	36
5	68	36
7	124	60
5	68	36
5	68	36
0	0	0

TASSWORD 2

BASICODE 3

BASICODE AMENT

Ga eens na of de getallen kloppen. De bitnummers staan er boven

Zoals we zien zijn er duidelijke verschillen. Tassword staat helemaal rechts in de matrix. Bas. 3 bijna in het midden en vrij breed. Versie Ament precies in het midden en smaller dan Bas.3. We kunnen Bas.3 en versie Ament heel gemakkelijk naar rechts schuiven door alle getallen te delen door 4. Beide zullen dan een plaats naar rechts in de matrix opschuiven. Maar Bas.3 zal dan toch te breed blijken voor de 64 kolommen van Tassword. De versie Ament past echter heel goed.

Zo kunnen we ook de charset van Tassword vermenigvuldigen met 8 en deze in Basicode gebruiken, maar het resultaat is niet optimaal.

Nog even in het kort hoe we dit kunnen doen. CLEAR 59999. laad de 42 koloms routine van Basicode 2 versie Ament. Tik de volgende regels in : 10 FOR k = 60074 TO 60841 : POKE k , PEEK k / 4 : NEXT k 20 SAVE "Ament" CODE 60074 , 768 : VERIFY " " CODE enter en RUN. Reset nu de computer en laad Tassword 2. Ga naar



Basic en tik in : LOAD "" CODE 61184, laad de charset en bekijk het resultaat.

Nu nog iets heel anders over Basicode 2 en 3. Beide versies beschikken n.l. over een 42 koloms routine en de versie Ament zelfs over een 51 koloms. Deze routines zijn soms heel goed te gebruiken in eigen programma's. Het is dan ook heel gemakkelijk om de routines er uit te lichten voor eigen gebruik. Het beeldscherm van de Spectrum beschikt dan over 42 (51) kolommen en tot 24 regels. Zelfs PRINT AT en TAB zijn dan aangepast zodat ook PRINT AT 23, 40; goed werken en ook TAB 40;.

De byzonderheden zijn als volgt :

Basicode 2 versie Ament :

42 koloms routine CODE 60000, 1611

CLOSE # 2: RANDOMIZE 2 : RANDOMIZE USR 60000 :

PRINT CHR\$ 1 ;: REM CHR\$ 2 VOOR 51 KOLOMMEN

PRINT CHR\$ 0; CHR\$ 23 ;: REM 23 VOOR AANTAL SCHERMREGELS

charset 60074, 768 of 896

Wilt U ook de beschikking over GOSUB 110 en GOSUB 120 neem dan ook de regels 110 en 120 over uit het Basicode vertaalprogramma.

Basicode 2 versie Breedenbeek

42 koloms routine CODE 57989, 1715

RANDOMIZE USR 57989

PRINT CHR\$ 5 ; CHR\$ 24 ; CHR\$ 24 ;

REM CHR\$ 5 SCROLL AAN

CHR\$ 4 SCROLL UIT

CHR\$ 24 + CHR\$ N = AANTAL SCHERMREGELS

Zie verder als boven

charset 58936, 768 of 896

RANDOMIZE USR 57993 REM 42 kolommen naar printer

RANDOMIZE USR 58048 REM 32 kolommen naar scherm

RANDOMIZE USR 58058 REM 32 kolommen naar printer

LET OP!! PRINTERSTURING WERKT NIET OP ALLE PRINTERS:

GEBRUIK DAN EIGEN PRINTERROUTINE.

Basicode 3 nieuwe versie

42 koloms routine CODE 56800, 2079

RANDOMIZE USR 56800

Verder geen controlecodes. 24 regels en 42 kolommen.

Zie verder als boven

charset 58112, 768 of 896

Met deze routine is het mogelijk om bij INPUT de cursor achter de PRINT instructie te krijgen d.m.v. INPUT # 2 ;. Er zal echter niets afgedrukt worden zoals bij de Spectrum of in Basicode. Ook de INKEY\$ routine kan eenvoudig overgenomen worden door de regels 200 e. v. in een eigen programma te zetten. Basicode 3 zet dan ook weer de cursor in het beeldscherm en de variabele IN krijgt de code van de ingedrukte toets. Zeer toepasselijk voor sturing van een menu.

Nadeel van deze routines is natuurlijk wel, dat RAMTOP verlaagd moet worden, bij de een wat meer dan de ander, maar het kost toch een flink stuk geheugen.

Tenslotte moet nog vermeld worden, dat het begin van het basicprogramma wordt verschoven. Dit begin is te vinden in de systeemvariabele PROG d.m.v. PEEK 23635 + 256 \* PEEK 23636.

Dit is het dan voor deze keer. Stoei er wat mee en steek er wat van op. Tot de volgende keer

drs. PC SEXZTRUM  
VOORLICHTER.





een melding gaat printen, want dan overschrijft hij je tekst. Een PAUZE 0 opdracht laat de tekst net zo lang staan totdat je een toets indrukt.

## \* WORTELS

De derde machts wortel van 27 krijg je door PRINT 27^(1/3) in te typen en de vijfde machts wortel van 7776 door PRINT 7776^(1/5). In het algemeen  $n^{(1/r)}$  geeft de r-de machts wortel van n. LET OP ^ betekent tot de macht.

## \* URIJE BYTES

PRNT 65535-USR 7962 geeft het aantal bytes dat nog vrij is. Dit geldt alleen voor de 48K Spectrum.

## \* GEHEUGENBESPARINGEN

- 1 Meer opdrachten op één regel plaatsen.
- 2 In een programma één keer LET a=1 gebruiken en verder overal waar het getal 1 nodig is de variable a gebruiken.
- 3 In plaats van PAPER 1: INK 6: FLASH 0: BRIGHT 1: CLS te gebruiken, type je in POKE 23693,78: CLS. Het getal 78 is een attribute waarde

## \* REGEL 0

Een regel 0 maak je door eerst regel 10 in te typen, daarna POKE 23755,0: POKE 23756,0: LIST. Een regel 0 kan niet zomaar gedelete worden. Je eigen regel 0, b.v. met je naam erin, kan je mergen met een eigengemaakt programma.

## \* MID\$, LEFT\$, RIGHT\$

MID\$(A\$,3,2) neemt twee karakters van A\$, te beginnen bij het derde karakter. Voor de spectrum is dat dus A\$(3 TO 5). LEFT\$(A\$,2) zijn de eerste twee karakters van A\$. Dus A\$(1 TO 2). RIGHT\$(A\$,2) zijn de laatste twee karakters van A\$. Dus A\$( LEN A\$-2 TO).

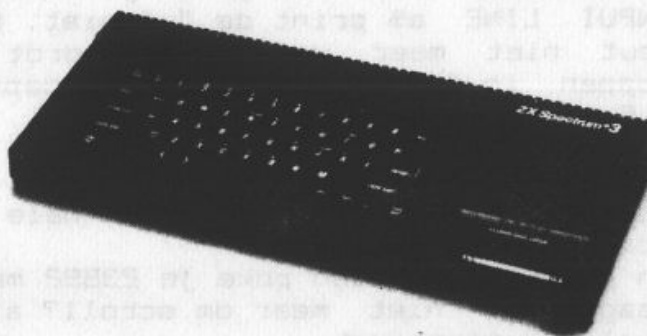
Ik hoop dat jullie iets aan deze tips hebben. Als er lezers zijn die zelf ook een tip hebben, stuur ze dan op naar de redactie, er kan een andere lezer zijn die al tijden naar de betreffende mogelijkheid zoekt.

Orm.

## NIEUWS OVER DE +3

**G**oed nieuws eindelijk voor de mensen die een +3 aan willen schaffen. Deze prachtige machine is in Engeland net in prijs verlaagd van 249 pond naar 199 pond!

Deze prijsverlaging werd overigens al lang verwacht door de computer-industrie, omdat werd verwacht dat Amstrad de +3 veel te duur op de markt had gebracht. Verder nieuws voor de +3 komt van Romantic Robot die inmiddels Multiface 3 hebben uitgebracht. Multiface 3 zet programma's op cassette over naar disk voor de 128K en de +3. Samen met de prijsverlaging voor de +3 kon dit wel eens een nieuwe hoop betekenen voor de Spectrum.





## S N E L & P R E C I E S R E K E N E N M E T D E Z X 81 (10)

Het onderwerp in kwestie is het snelle  
GENEREREN VAN PRIEMGETALLEN

De vorige keer behandelde ik het programma dat tot 65536 (2\*\*16) deze opgave uitvoert, in dit artikel wordt deze toepassing tot 1000000 (1E6) besproken. Het eerste programma heb ik genoemd PRIMEGENERATOR afgekort PRIME, het tweede PRIMEGENERATOR afgekort PRIME.

### TOEVOEGINGEN

Voor deze uitbreiding zijn een aantal toevoegingen nodig, met name:

- Een C(oun)T(e)R is ingevoerd. De 78498 PRIEMgetallen < 1E6 die er blijken te zijn zouden voor de storage een geheugenkapaciteit van bijna 314000 bytes vergen. De CTR maakt een systematische beperking van de opslag mogelijk.

- Het te onderzoeken getal NR (NUMBER) bestaat nu 3 bytes. Bij gebrek aan registers moet voor de opslag van 1 byte (HI-NR) een systeemvariabele (4048) worden gebruikt.

- De in het voorgaande artikel reeds genoemde DIVISOR-LIST die de delers tot uitsluitend PRIEMgetallen beperkt. In deze tabel staan de 5de t/m 169ste PRIEM, respectievelijk 0B (11) en 03F1 (1009).

- In plaats van directe PRINT-OUT van de 3-bytes-grote hex getallen worden de 5 digits hex resultaten eerst geconverteerd in 6 digits decimaal, terwijl dan tevens het rangnummer (RANK) berekend en opgeslagen wordt. Deze gegevens, 4 bytes per combinatie, worden gePRINT, de RANK in inverse characters.

FIG. 3A PRIMEGENERATOR <1E6

```
1 REM-REGEL 2 + 158 + 97 + 3
+ 32 + 48 + 330 = 670 BYTES
3 RAND USR 16516
4 RAND USR 16674
5 RAND USR 16774
```

### PRIMEGENERATOR

Het BASIC-programma, figuur 3A, bestaat uit de REM-regel van 670 bytes. MC-routines en tabellen die hieronder worden besproken, gevolgd door 3 regels met achtereenvolgens de aanroep voor het rekenen (regel 3) daarna die voor de conversie (regel 4) en tenslotte die voor het op het scherm zichtbaar maken van de resultaten (regel 5).

Na RUN en het beëindigen van het rekenwerk stopt het programma met REPORT 9/3, zodat voor de conversie de hex resultaten kunnen worden bekeken. Met CONT volgt dan de uitvoering van regel 4 en wordt de eerste pagina met uitkomsten gePRINT. Hierna kunnen met CONT de volgende pagina's zichtbaar worden gemaakt.

Desgewenst kan pagina na pagina met COPY naar een printer worden gevoerd.

Het rekenen moet in FAST gebeuren. De conversie kan ook in SLOW, maar kost dan onnodig veel tijd. PRINTen in SLOW is niet bezwaarlijk.

De HEX-HEX en DEC-DEC DUMPS van figuur 3B geven achtereenvolgens de REKEN/CONVERSIE/PRINT-RTNS de TABLE en de DIVISOR-LIST.

FIG. 3B PRIMEGENERATOR HEXDUMP

```
4082--7676 110B 0021 0047--170
408A--227B 4022 4640 083E--1CB
4092--5F08 0EA6 0641 79EE--2C9
409A--D828 F70A 835F 3006--317
40A2--1420 03FD 3448 0C21--100
40AA--D641 D57E 2346 23C5--38B
40B2--D9D1 5FC1 3A48 4021--3AD
40BA--0000 D906 18D9 CB11--2AC
40C2--CB10 17ED 6AED 5230--3B8
40CA--0119 D910 F0D9 7CB5--3FD
40D2--D928 C1D9 7A4B 2100--381
40DA--0006 101F CB19 3001--14A
40E2--19CB 1CCB 1D10 F41F--30B
40EA--CB19 677D D9FD 4648--42C
40F2--9038 B720 09D5 D901--427
40FA--69ED 52D9 38AC 0830--3AA
4102--200F 3E64 2A46 4023--1A4
410A--7323 7223 7023 2246--226
4112--4008 3E0F 9020 0421--16A
411A--D1BD 19D2 9640 CF08--426
4122--1100 476B 62E5 0104--20F
412A--B909 CB3C CB1D 083C--3B8
4132--CB1D CD73 41E1 234E--3BB
413A--2346 237E 23E5 F559--370
4142--60CD 6741 C104 0528--2C7
414A--14A7 2185 410E 031B--1CE
4152--1A8E 2712 2D0D 20F7--232
415A--1313 1310 EC21 C0AC--2C2
4162--19E1 30C1 C901 1027--2EC
416A--CD76 4101 6400 CD76--32C
4172--4101 0100 3E99 3C27--17D
417A--A7ED 4230 F909 1213--32D
4182--C906 5536 2A7B 4070--2BC
418A--E603 0E1C 2002 0E9C--1DF
4192--7E1F 1F1F 1FE6 0F81--270
419A--D77E E60F 81D7 2322--3E7
41A2--7B40 18E3 --1B6
```

```
41A6--0204 0204 0602 0604--01E
41AE--0204 0606 0206 0402--020
41B6--0604 0608 0402 0402--024
41BE--0408 0604 0602 0406--028
41CE--0206 0604 0204 0602--020
41DE--0604 0204 020A 020A--028
```

```
41D6--0B00 0D00 1100 1300--03C
41DE--1700 1D00 1F00 2500--078
41EE--2900 2B00 2F00 3500--0B8
41FE--3B00 3D00 4300 4700--102
41F6--4900 4F00 5300 5900--144
41FE--6100 6500 6700 6B00--198
4206--6D00 7100 7F00 8300--1E0
420E--8900 8B00 9500 9700--240
4216--9D00 A300 A700 AD00--294
421E--B300 B500 BF00 C100--2E8
4226--C500 C700 D300 DF00--33E
422E--E300 E500 E900 EF00--3A0
4236--F100 FB00 0101 0701--1F6
423E--0D01 0F01 1501 1901--04E
4246--1B01 2501 3301 3701--0AE
424E--3901 3D01 4B01 5101--116
4256--5B01 5D01 6101 6701--184
425E--6F01 7501 7B01 7F01--1E2
4266--8501 8D01 9101 9901--240
426E--A301 A501 AF01 B101--2AC
```

# BULLETIN SGG

```

4276--B701 BB01 C101 C901--300
427E--CD01 CF01 D301 DF01--352
4286--E701 EB01 F301 F701--300
428E--FD01 0902 0B02 1002--135
4296--2302 2002 3302 3902--004
429E--3B02 4102 4B02 5102--120
42A6--5702 5902 5F02 6502--17C
42AE--6902 6B02 7702 8102--104
42B6--8302 8702 8D02 9302--232
42BE--9502 A102 A502 AB02--28E
42C6--B302 BD02 C502 CF02--30C
42CE--D702 DD02 E302 E702--386
42D6--EF02 F502 F902 0103--2E7
42DE--0503 1303 1003 2903--05A
42EE--2B03 3503 3703 3B03--0DE
42FE--3D03 4703 5503 5903--13E
42F6--5B03 5F03 6D03 7103--1A4
42FE--7303 7703 8B03 8F03--210
4306--9703 A103 A903 AD03--29A
430E--B303 B903 C703 CB03--30A
4316--D103 D703 DF03 E503--378
431E--F103 --0F4

```

```

16514-7676 110B 0021 0047- 358
16522-227B 4022 4640 083E- 459
16530-5F08 0EAB 0641 79EE- 713
16538-0628 F70A 835F 3006- 791
16546-1420 03FD 3448 0C21- 477
16554-0641 D57E 2346 23C5- 955
16562-09D1 5FC1 3A48 4021- 941
16570-0000 D906 1B09 CB11- 684
16578-CB10 17ED 6AED 5230- 952
16586-0119 D910 F0D9 7CB5-1021
16594-0928 C109 7A48 2100- 897
16602-0006 101F CB19 3001- 330
16610-19CB 1CCB 1010 F41F- 779
16618-CB19 677D D9FD 4648-1068
16626-9038 B720 09D5 D9D1-1063
16634-69ED 52D9 33AC 083D- 938
16642-200F 3E64 2A46 4023- 420
16650-7323 7223 7023 2246- 550
16658-4008 3E0F 9020 0421- 362
16666-01B0 19D2 9540 CF08-1062
16674-1100 476B 62E5 0104- 527
16682-B909 CB3C CB10 CB3C- 952
16690-CB1D CD73 41E1 234E- 955
16698-2346 237E 23E5 F569- 880
16706-60CD 6741 C104 0528- 711
16714-14A7 2185 410E 031B- 462
16722-1A8E 2712 20D0 20F7- 562
16730-1313 1310 EC21 C0AC- 706
16738-19E1 30C1 C901 1027- 748
16746-CD76 4101 6400 CD76- 812
16754-4101 0100 3E99 3027- 381
16762-A7ED 4230 F909 1213- 813
16770-C906 5536 2A7B 407D- 700
16778-E603 0E1C 2002 0E9C- 479
16786-7E1F 1F1F 1FE6 0F81- 624
16794-D77E E60F 81D7 2322- 999
16802-7B40 18E3 - 438

```

```

16806-0204 0204 0602 0604- 030
16814-0204 0606 0206 0402- 032
16822-0504 0608 0402 0402- 036
16830-0408 0604 0602 0406- 040
16838-0206 0604 0204 0602- 032
16846-0504 0204 020A 020A- 040

16854-0B00 0D00 1100 1300- 060
16862-1700 1D00 1F00 2500- 120
16870-2900 2B00 2F00 3500- 184
16878-3B00 3D00 4300 4700- 258
16886-4900 4F00 5300 5900- 324
16894-6100 6500 6700 6B00- 408
16902-6D00 7100 7F00 8300- 480
16910-8900 8B00 9500 9700- 576
16918-9D00 A300 A700 A000- 660
16926-B300 B500 BF00 C100- 744
16934-C500 C700 D300 DF00- 830
16942-E300 E500 E900 EF00- 928
16950-F100 FB00 0101 0701- 502
16958-0D01 0F01 1501 1901- 078
16966-1B01 2501 3301 3701- 174
16974-3901 3D01 4B01 5101- 278
16982-5B01 5D01 6101 6701- 388
16990-6F01 7501 7B01 7F01- 482
16998-8501 8D01 9101 9901- 576
17006-A301 A501 AF01 B101- 684
17014-B701 BB01 C101 C901- 768
17022-CD01 CF01 D301 DF01- 850
17030-E701 EB01 F301 F701- 960
17038-FD01 0902 0B02 1002- 309
17046-2302 2D02 3302 3902- 196
17054-3B02 4102 4B02 5102- 288
17062-5702 5902 5F02 6502- 380
17070-6902 6B02 7702 8102- 468
17078-8302 8702 8D02 9302- 562
17086-9502 A102 A502 AB02- 654
17094-B302 BD02 C502 CF02- 780
17102-D702 DD02 E302 E702- 902
17110-EF02 F502 F902 0103- 743
17118-0503 1303 1003 2903- 106
17126-2B03 3503 3703 3B03- 222
17134-3D03 4703 5503 5903- 318
17142-5B03 5F03 6D03 7103- 420
17150-7303 7703 8B03 8F03- 528
17158-9703 A103 A903 AD03- 666
17166-B303 B903 C703 CB03- 778
17174-D103 D703 DF03 E503- 888
17182-F103 -- 244

```

MACHINECODEROUTINE FIG, 3C  
 Deze routine bestaat uit 670  
 bytes en wel 2 bytes hexcode 76  
 + 158 bytes REKEN-RTN + 97 bytes  
 CONVERSIE-RTN + 3 bytes MULTI-  
 PLICAND HI-NR + 32 bytes PRINT-  
 RTN + 48 bytes TABLE + 330 bytes  
 DIVISOR-LIST.

FIG. 3C PRIMEGENERATOR MC-RTN

4082--76/76

HALT/HAUT

INITIATE

```

INIT 4084--110B00
      4087--210047
      408A--227B40
      408D--224640
      4090--00
      4091--3E5F
      4093--00

```

```

LD DE,000B S. PRIME : 11 d
LD HL,4700 STORE-BEGIN
LD (407B),HL PTR-PRINTOUT
LD (4046),HL STORE-PTR
EX AF,AF" A1 = CTR
LD A,5F SET CTR 95 d
EX AF,AF"

```

MAKE NUMBER

```

NXT-POUND 4094--0EAB
NXT-NR 4096--0641
      4098--79
      4099--EED6
      409B--28F7

```

```

LD C,A6 LD-TABLE
LD B,41 HI-TABLE
LD A,C
XOR D6 LD-TABLE-END +1
JR Z,4094 NXT-POUND

```



# BULLETIN SIGG

	409D--0A	LD	A, (BC)	
	409E--83	ADD	A, E	
	409F--5F	LD	E, A	DE = LO-NR
	40A0--3006	JR	NC, 40A8	INC-POS
	40A2--14	INC	D	
	40A3--2003	JR	NZ, 40A8	INC-POS
	40A5--FD3448	INC	(IY+48)	HI-NR
INC-POS	40A8--0C	INC	C	C = NXT-POS in TABLE

## NXT-DIVISOR/PREPARE + DIVISION

NXT-DIVISOR	40A9--21D641	LD	HL, 41D6	BEGIN-LIST
	40AC--D5	PUSH	DE	stack: LO-NR
	40AD--7E	LD	A, (HL)	LSB-DIVISOR
	40AE--23	INC	HL	
	40AF--46	LD	B, (HL)	MSB-DIVISOR
	40B0--23	INC	HL	HL = NXT-POS in LIST
	40B1--C5	PUSH	BC	stack: MSB-DIVISOR
	40B2--D9	EXX		
	40B3--D1	POP	DE	D' = MSB-DIVISOR
	40B4--5F	LD	E, A	D'E' = DIVISOR
	40B5--C1	POP	BC	B'C' = LO-NR
	40B6--3A4840	LD	A, (4048)	A = HI-NR
	40B9--210000	LD	HL, 0000	RES REMAINDER
	40BC--D9	EXX		
	40BD--0616	LD	B, 16	24 BITS SHIFT
				if carry : not relevant
DIV-LOOP	40BF--D9	EXX		
	40C0--CB11	RL	C	
	40C2--CB10	RL	B	A!, B'C' = NR
	40C4--17	RLA		
	40C5--ED6A	ADC	HL, HL	-> no carry
	40C7--ED52	SBC	HL, DE	TRIAL-SUBTRACT
	40C9--3001	JR	NC, 40CC	NO-PESTORE
	40CB--19	ADD	HL, DE	RESTORE HL
NO-RESTORE	40CC--D9	EXX		
	40CD--10F0	DJNZ	40BF	DIV-LOOP
	40CF--D9	EXX		
	40D0--7C	LD	A, H	
	40D1--B5	OR	L	NO REMAINDER?
	40D2--D9	EXX		
	40D3--28C1	JR	Z, 4096	NXT-NR if so

## DIVISOR TESTED

DIVISOR-GRD	40D5--D9	EXX		
	40D6--7A	LD	A, D	D'E' = MULTIPLICAND
	40D7--4B	LD	C, E	A, C' = MULTIPLIER
	40D8--210000	LD	HL, 0000	RES HI-PRODUCT
	40DB--0610	LD	B, 10	16 BITS SHIFT
MULT-LOOP	40DD--1F	RRA		
	40DE--CB19	RR	C	
	40E0--3001	JR	NC, 40E3	NO-ADD
	40E2--19	ADD	HL, DE	
NO-ADD	40E3--CB1C	RR	H	H' = 00 finally, so
	40E5--CB1D	RR	L	L' = HI-PRODUCT
	40E7--10F4	DJNZ	40DD	MULT-LOOP
	40E9--1F	RRA		Final SHIFT of
	40EA--CB19	RR	C	A, C' -> LO-PRODUCT
	40EC--67	LD	H, A	H', C' = LO-PRODUCT
	40ED--7D	LD	A, L	A = HI-PRODUCT
	40EE--D9	EXX		
< NUMBER?	40EF--FD4648	LD	B, (IY+48)	B = HI-NR
	40F2--90	SUB	B	A-B : see text
	40F3--38B7	JR	C, 40AC	NXT-DIVISOR
	40F5--2009	JR	NZ, 4100	PRIME
	40F7--D5	PUSH	DE	stack: LO-NR
	40F8--D9	EXX		
	40F9--D1	POP	DE	D'E' = LO-NR
	40FA--69	LD	L, C	H'L' = LO-PRODUCT
	40FB--ED52	SBC	HL, DE	see text
	40FD--D9	EXX		
	40FE--38AC	JR	C, 40AC	NXT-DIVISOR if
				PRODUCT < NUMBER

## COUNT/WRITE PRIME + TEST STOP

PRIME	4100--08	EX	AF, AF"	
	4101--3D	DEC	A	A' = CTR
	4102--200F	JR	NZ, 4113	NO-STORAGE
	4104--3E64	LD	A, 64	SET CTR at 100 d
	4106--2A4640	LD	HL, (4046)	STORE-CTR
	4109--23	INC	HL	
	410A--73	LD	(HL), E	LSB LO-NR
	410B--23	INC	HL	

# BULLETIN SGG

	410C--72	LD	(HL), D	HSB LO-NR
	410D--23	INC	HL	
	410E--70	LD	(HL), B	HI-NR
	410F--23	INC	HL	
NO-STORAGE	4110--224540	LD	(4045), HL	NXT-STORE-POS
	4113--08	EX	AF, AF"	
TEST-STOP	4114--3E0F	LD	A, 0F	A = LIMIT HI-NR
	4116--90	SUB	B	B = HI-NR ; see text
	4117--2004	JR	NZ, 411D	TO-NXT-NR
	4119--21D1BD	LD	HL, 8DD1	NEG-LIMIT LO-NR
	411C--19	ADD	HL, DE	see text
TO-NXT-NR	411D--D29640	JP	NC, 4096	NXT-NR
	4120--CF08	RST	08; 9	RETURN to BASIC
				and STOP

## RANK + CONVERT HEX-TO-DEC + TEST

16674	4122--110047	LD	DE, 4700	STORE-BEGIN
	4125--6B	LD	L, E	DE = WRITE-POS
NEXT	4126--62	LD	H, D	HL = STORE PTR
	4127--E5	PUSH	HL	stack
WRITE-RANK	4128--0104B9	LD	BC, 8904	NEG STORE-BEGIN +4
	4128--09	ADD	HL, BC	HL = relative STORE-POS
	412C--CB3C	SRL	H	
	412E--CB1D	RR	L	divide
	4130--CB3C	SRL	H	
	4132--CB1D	RR	L	by 4
	4134--CD7341	CALL	4173	WRITE-2
CONVERT	4137--E1	POP	HL	retrieve STORE-PTR
	4138--23	INC	HL	
	4139--4E	LD	C, (HL)	LSB LO-NR
	413A--23	INC	HL	
	413B--46	LD	B, (HL)	HSB LO-NR
	413C--23	INC	HL	
	413D--7E	LD	A, (HL)	HI-NR
	413E--23	INC	HL	HL = NXT-POS
	413F--E5	PUSH	HL	of STORE-PTR
	4140--F5	PUSH	AF	stack HI-NR
	4141--09	LD	L, C	
	4142--60	LD	H, B	HL = LO-NR
	4143--CD6741	CALL	4167	WRITE-6
	4146--C1	POP	BC	B = HIJNP
	4147--04	INC	BC	
	4148--05	DEC	BC	B = 00?
	4149--2814	JR	NZ, 415F	TEST-POS if yes
WRITE-ADD	414B--A7	AND	A, 415F	
	414C--218541	LD	HL, 4185	LO-DIG MULTIPLICAND
	414F--0E03	LD	C, 03	3 bytes
NXT-2 DIGS	4151--15	DEC	DE	decrement WRITE-POS
	4152--1A	LD	A, (DE)	
	4153--8E	ADC	A, (HL)	add 2 DIGS
	4154--27	DAA		decimally
	4155--12	LD	(DE), A	WRITE
	4156--2D	DEC	L	L = PTR-MULTIPLICAND
	4157--0D	DEC	C	nxt-lower byte
	4158--20F7	JR	NZ, 4151	NXT-2 DIGS
	415A--13	INC	DE	
	415B--13	INC	DE	
	415C--13	INC	DE	
	415D--10EC	DJNZ	414B	DE = WRITE-POS RANK
TEST-POS	415F--21C0AC	LD	HL, ACC0	NXT-ADD
	4162--19	ADD	HL, DE	NEG LAST-POS
	4163--E1	POP	HL	
	4164--30C1	JR	NC, 4127	retrieve STORE-PTR
	4166--C9	RET		NEXT

## WRITE DECIMAL DIGITS

WRITE-6	4167--011027	LD	BC, 2710	10.000 d
	416A--CD7641	CALL	4176	COUNT 2 DIGS
	416D--016400	LD	BC, 0064	100 d
	4170--CD7641	CALL	4176	COUNT 2 DIGS
WRITE-2	4173--010100	LD	BC, 0001	1 d
COUNT 2 DIGS	4176--3E99	LD	A, 99	SET CTR at -1
CTR-LOOP	4178--3C	INC	A	
	4179--27	DAA		A = DECIMAL-CTR
	417A--A7	AND	A	
	417B--ED42	SBC	HL, BC	
	417D--30F9	JR	NC, 4178	CTR-LOOP
	417F--09	ADD	HL, BC	add BC back
	4180--12	LD	(DE), A	WRITE 2 DIGS
	4181--13	INC	DE	NXT-WRITE-POS
	4182--C9	RET		

4183--065536 MULTIPLICAND HI-NR



# BULLETIN SIG

## PRINTOUT INVERSE RANK AND PRIME

16774	4186--2A7B40	LD	HL, (407B)	PTR-PRINTOUT
NXT	4189--7D	LD	A, L	
	418A--E603	AND	03	I = 00:04/08/0C ?
	418C--0E1C	LD	C, 1C	NORMAL CHR
	418E--2002	JR	NZ, 4192	NO INVERSE if not
	4190--0E9C	LD	C, 9C	INVERSE CHR
NO-INVERSE	4192--7E	LD	A, (HL)	
	4193--1F	RRR		
	4194--1F	RRR		
	4195--1F	RRR		
	4196--1F	RRR		
	4197--E60F	AND	0F	MAKE CHR
	4199--81	ADD	A, C	PRINT LEFT-DIG
	419A--D7	RST	10	
	419B--7E	LD	A, (HL)	
	419C--E60F	AND	0F	MAKE CHR
	419E--81	ADD	A, C	PRINT RIGHT-DIG
	419F--D7	RST	10	
	41A0--23	INC	HL	
	41A1--2A7B40	LD	(407B), HL	PTR-PRINTOUT
	41A4--18E3	JR	4189	NXT

## TABLE

TABLE-BEGIN	41A6: SEE HEXDUMP 6*8 BYTES	
	.....41D5	TABLE-END

## DIVISOR-LIST

BEGIN-LIST	41D6: SEE HEXDUMP 41*8 BYTES +
	431E--F103

De opslagplaats voor de berekende PRIEMgetallen -STORE- begint op adres 4700h. Zoals al gezegd worden 4 bytes per combinatie RANK/PRIEMgetal gebruikt. Indien gewenst kan RAMTOP naar STORE-BEGIN worden verplaatst waardoor in een 16K geheugen 3900h=14592d bytes ter beschikking staan, genoeg voor 3648 combinaties. In de voorbeelden die ik later geef worden maximaal 3000 combinaties opgeslagen en is die verplaatsing dus niet nodig.

De routine begint met 2 bytes 76 om het bekende gevaar van een REM-regel die meer dan 1 pagina op het scherm inneemt te vermijden. Het rekenen wordt gestart bij INIT op adres 4084h=16516d met het SETten van DE het register voor de eerste twee bytes van het te onderzoeken getal NR. DE, LO-NR begint met 11d, de 5de PRIEM. Zoals eerder uiteengezet correspondeert dit getal met het eerste element uit de zeef tabel: TABLE-BEGIN, adres 41A6. Vorige keer is besproken dat de cyclus van deze tabel met 48 elementen  $2*3*5*7 = 210$  is. In verband met de DIVISOR-LIST die t/m 1009 gaat, zouden we met dat getal kunnen starten en de daarbij behorende waarde van de TABLE-pointer C. Of zonder meer met als LO-NR:  $11 + 4*210 = 851$ . Ik heb dat niet gedaan, omdat de rekentijd voor de PRIEMgetallen van 13 -het eerste dat berekend wordt- t/m 1009 slechts 1 seconde is!

In INITIATE worden vervolgens zowel de P(oin)T(e)R voor de later volgende PRINTOUT als de STORE-PTR op het beginadres van STORE geSET.

Als CTR is A' gebruikt en wel wordt geteld van hoog naar laag, zodat nul betekent "dat is 'm'".

Uitgaande van opslag van de berekende PRIEMgetallen per 100 (zie adres 4104) moeten we A' dus SETten op  $100 - 5 = 5F$ .

MAKE NUMBER maakt gebruik van de zeef tabel TABLE op de de vorige keer besproken wijze. DE bevat LO-NR en iedere keer als D = 0 wordt verhoogd we  $(4048) = HI-NR$  met 1. Deze systeemvariabele ligt in het PRBUFF-gebied en is daarom automatisch op nul geSET als we de routine starten. C fungeert weer, samen met de B die geSET wordt op 41h, als pointer voor TABLE.

De (NXT)-DIVISOR komt als gezegd uit de DIVISOR-LIST. HL vormt de PTR voor deze tabel en behoeft door het veelvuldig gebruik maken van de exchange registers (via de instructie EXX) niet te worden gestacked. De DIVISOR wordt uitgelezen en overgebracht naar D'E' voor de DIVISION-LOOP. NR wordt voor die LOOP omgezet van  $(4048)$ , DE naar A,B'C'. De waarden in de eerstgenoemde "registers" blijven behouden voor volgende ronden.

De DIVISION-RTN werkt identiek aan die behandeld in PRIME, we hebben nu echter te maken met 3 bytes, dus 24 bits: B = 18h. En we hebben een register meer in gebruik: B', daarom moet DJNZ buiten de EXX-RTN blijven.

De DIVISOR D'E' wordt gekwadreteerd: 16 bits voor de MULT-LOOP en daarom kan TEST-1 van PRIME vervallen. Wat blijft is de TEST: < NUMBER? En die moet nu worden uitgevoerd voor 3 bytes. Eerst wordt A = HI-PRODUCT vergeleken met B =  $(4048) = HI-NR$ . Er zijn 3 mogelijkheden als resultaat van SUB B, adres 40F2:

. A < B, het deelbaarheidsonderzoek moet worden voortgezet: via JR C naar NXT-DIVISOR.  
 . A > B, we hebben een PRIEM gevonden: de carry is 0 en via JR NZ gaan we naar het volgende deel van de RTN met label PRIME.  
 . A = B, de HI-waarden zijn gelijk en het onderzoek van de LO-waarde moet uitsluitend geven over het TEST-resultaat; daartoe wordt DE = LO-NR naar D'E' overgebracht en vergeleken met LO-PRODUCT dat van A,C' naar H'L' is overgebracht.

In 't laatste deel van de rekenroutine wordt het NR dat gebreken is een PRIEM te zijn geteld. En opgeslagen in de STORE indien de CTR A' nul is. GESTORED worden drie bytes van laag naar hoog: E, D en B = (4048).  
 Vervolgens wordt in TEST-STOP NR vergeleken met de grenswaarde LIMIT, vastgelegd in A (HI) en in HL het komplement van de LO-waarde. A > B levert niet alleen NZ op, maar ook NC en zo wordt via JP NC, NXT-NR het onderzoek voortgezet. Als A = B beslist de waarde van LO-NR, getest op de allengs bekende wijze, wat er gebeurt.

Als de grenswaarde is bereikt wordt teruggekeerd naar BASIC en STOPT het programma. Deze grenswaarde is in figuur 3C op de adressen 4114 en 4119 zodanig gekozen dat GESTORED wordt als de grootste PRIEM < 1E6 is bereikt. Deze PRIEM is 999983 = 15\*65536 + 16943. Het komplement van dat laatste getal is in hex BDD1.

LET OP: als bij een LIMIT HI-NR van 0F de NEG-LIMIT LO-NR < BDD1 zou zijn gekozen gaat de routine "eeuwig" door.

De CONVERSION-RTN SET de WRITE-POSITION (DE) en de STORE-PTR (HL) op 4700h, STORE-BEGIN. Dan wordt de relatieve STORE-POS bepaald door het komplement van STORE-BEGIN, vermeerderd met 04, bij HL op te tellen. Waarom die +04? Op adres 4701/03 staat het eerste GESTORED PRIEMgetal. Op 4700 moet dus de RANK 01 worden geschreven. STORE-POS wordt door 4 gedeeld en geeft dan de gewenste waarden 01, 02, enz. op het juiste adres.

LET WEL: om de STORE-kapaciteit optimaal te benutten en de RANK eenvoudig te kunnen bepalen en 4 combinaties RANK/PRIEM per regel te kunnen PRINTEN gebruik ik slechts 4 bytes per combinatie, waarvan dus maar 1 voor de RANK beschikbaar is. Dat -decimale-rangnummer loopt derhalve van 01 t/m 99 en gaat dan door met 00. De honderdtallen volgen echter uit het STORE-adres.

Deze RANK wordt opgeslagen op de adressen 4700, 4704 enz met behulp van de subroutine WRITE DECIMAL DIGITS, adres 4173 label WRITE-2. Voor de goede orde: op dat adres moet ook de waarde van B geSET zijn, zodat niet kan worden volstaan met LD C, 01.

Vervolgens worden de 3 bytes van het getal uitgelezen naar C, B en A. Eerst wordt BC geconverteerd en decimaal (3 bytes) ge-

schreven op DEZELFDE adressen als waar zoeven is gelezen. En wel omgekeerd aan de hex opslagmethode, dus de laagste 2 digits op het hoogste adres. Dit gaat via de ingang WRITE-6 van de subroutine. De volgende stap is de MULTIPLICAND 65536, opgeslagen op Adres 4103/5 decimaal te vermenigvuldigen met HI-NR en op te tellen bij het zoeven geschreven getal. Dat gebeurt in het RTN-gedeelte AD 414B/5E. Maar niet dan nadat dit vermeden is voor HI-NR = 0: in een DJNZ-loop betekent B = 0 immers dat de loop 256-maal wordt doorlopen!

In TEST-POS wordt op de gebruikelijke manier door optelling van het komplement van de LAST-POS, de laatste positie die we willen omverteren, onderzocht of we moeten doorgaan of klaar zijn. De NEG LAST-POS AD 415F is als volgt bepaald: er zijn 784 combinaties van 4 bytes in de STORE als de routine klaar is en ACC0 is het komplement van 4\*784d = 0C40 + 4700h = 5340h.

WRITE DECIMAL DIGITS is geschikt voor maximaal het getal 9999999 door de trucs die zijn toegepast. Normaal wordt geteld per digit, maar dan kom je niet verder dan 99999 omdat de grootste deler 10000 (2710h) is. Dankzij DAA kunnen we 2 digits tegelijk bepalen. Maar dan zou de vereiste 'add A back' na de CTR-LOOP met AND A/DEC A/DAA moeten, omdat een carry DEC A/DAA (evenals natuurlijk INC A/DAA) fout laat functioneren. Probeer het maar!

Korter en sneller lossen we dit probleem op door het tellen niet te beginnen op 00 maar op 99 = -01, AD 4176. Deze waarde van A is tevens de enige waarbij INC A/DAA wel het korrekte resultaat geeft, ook al is er een carry. Hetzelfde geldt voor A = 00 bij DEC A/DAA.

PRINTOUT INVERSE RANK AND PRIME  
 Voor deze RTN is de PTR geSET op 4700 in INITIATE, maar we kunnen vanzelfsprekend iedere andere waarde kiezen. Waarbij we ons wel moeten realiseren, dat de CHRS die INVERSE worden GEPRINT zijn gebonden aan de in de toelichting bij de RTN vermelde waarden van L. Bij het uitprinten van (gedeelten van) STORE moeten we in ieder geval steeds beginnen op een adres dat een viervoud is, uitgaande van STORE-BEGIN. Maar als deze RTN voor andere programma's wordt gebruikt kunnen de data anders gegroepeerd zijn; die moeten dan eerst worden gerelokateerd.

De RTN zelf behoeft geen nadere toelichting. Per pagina worden 22\*4 = 88 getallen met rangnummer GEPRINT. De reden waarom dat laatste invers gebeurt zal duidelijk zijn: er is immers geen plaats voor spaties. Met CONT worden volgende pagina's zichtbaar gemaakt, omdat voor deze RTN een aparte aanroepregel is gebruikt.

Figuur 3C eindigt met de TABLE van 48 elementen en de DIVISOR-LIST met 165 stuks 2-bytes-grote



# BULLETIN SIGG

PRIEMgetallen. Zie hiervoor de bespreking van de TOEVOEGINGEN.

20 uitkomsten voor en na conversie en gePRINT. Ook de PRINTOUT van de laatste 10 regels.

## RESULTATEN

Zoals bij de behandeling van de RTNS is duidelijk geworden, is op een aantal adressen de gekozen vaste waarde van het register bepalend voor wat de RTN uiteindelijk berekent en vastlegt. In figuur 3C is die keuze -konform het onderwerp van dit artikel- afgestemd op het genereren van alle PRIEMgetallen die < 1E6 zijn. En ter beperking van de opslagruimte is gekozen voor vastlegging van iedere 100ste PRIEM. Er zijn vanzelfsprekend varianten door modificatie van die "vaste" waarden mogelijk. Een 5-tal daarvan staat in het onderstaande overzicht met de te kiezen waarden van de registers op de 7 relevante adressen. Enkele resultaten worden gegeven: het aantal opgeslagen PRIEMgetallen en het laatste. Verder de benodigde tijd in seconden voor de berekening (BRKNG) en voor de CONVERSIE.

De "titels" van de varianten spreken voor zichzelf. De 4de variant is de in dit artikel behandelde. In wezen zou hiervoor HL op AD 4119 iets groter en wel C35F kunnen zijn, korresponderend met het laatste getal 998561. Dat scheelt 22 s rekentijd, 0.2%.

regis- ter/AD	1E3	3E3	1E4	als slot	
	per1	per1	per1	FIG 3C<1E6	
DE 4084	000B	000B	000B	000B	3CA1
HL 4087	4714	4714	4700	4700	4700
A/4091	01	01	05	5F	01
A/4104	01	01	0A	64	01
A/4114	00	00	01	0F	0F
HL 4119	E111	94C7	66E7	BDD1	BDD1
HL 415F	A960	8A20	A960	ACC0	
aantal	1000	3000	1000	784	98
tijd(s)					
BRKNG	<18	96	575	11750	22
CONU	<8	59	<8	<6	
laatste getal	27449	998561			
	7919	104729	999983		

Er is geen merkbaar verschil in rekentijd of een PRIEM wordt opgeslagen of niet: "per 1" en "per 100" kosten evenveel tijd. De CONVERSIE-tijd is wel sterk afhankelijk van het aantal, zoals uit het overzicht blijkt. In het de vorige keer behandelde programma PRIME konden we komen t/m de 6542ste PRIEM. De rekentijd van dat PG, voor die varianten die vergelijkbaar zijn, is precies de helft van die van PRIME.

VOORBEELDEN van de resultaten van de behandelde routine worden gegeven in figuur 4A. En wel

FIG. 4A VOORBEELD RESULTATEN PROGRAMMA PER 100

PRIMES NR 91(00) T/M 110(00)  
NA DE BEREKENING:

```

4868--0080 7001 00D1 7401--244
4870--0045 7901 0015 7E01--153
4876--0083 8201 000D 8701--198
4880--00A1 8801 00AD 8F01--26A
4888--00A7 9401 0019 9901--1EF
4890--00D7 9D01 0019 A201--231
4898--00E7 A601 004F AB01--289
48A0--0005 B001 00A5 B401--210
48A8--0053 B901 00C1 BD01--28C
48B0--0079 C201 00DF C601--2E2

```

IDEM NA CONVERSIE:

```

4868--9109 4349 9209 5441--256
4870--9309 6581 9409 7813--2AA
4876--9509 8947 9610 0109--21E
4880--9710 1281 9810 2317--21C
4888--9910 3591 0010 4729--1EF
4890--0110 5943 0210 7033--162
4898--0310 8263 0410 9391--230
48A0--0511 0597 0611 1781--161
48A8--0711 2979 0811 4113--127
48B0--0911 5321 1011 6447--15A

```

PRINTOUT VAN DEZE 20 PRIMES

```

91094349980954418309658184097813
95098947981001090710128193102317
99103591001047290110594303107033
03108263041093910511059705111781
07112979061141130911532110116447

```

LAATSTE 10 REGELS PRINTOUT:  
PRIMES NR 745(00) T/M 764(00)

```

45944161459456474794702748948449
49949951509511615195264958954043
53955391549569535595836156959759
57961123589624415996375160965113
61966463609679616396934364970721
65971951689732896697473768976039
69977593709789477198049170981887
73983299749847017598610176987433
77988901789903237999169380993107
81994501839959098399720781998561

```

FIG. 4B VOORBEELDEN PRINTOUT PG-MODIFICATIE: "PER 1"

EERSTE 10 REGELS

```

05000013070000170800001909000023
10000029110000311200003713000041
14000043150000471600005317000059
18000061190000672000007121000073
22000079230000832400008925000097
26000101270001032800010729000109
30000113310001273200013133000137
34000139350001493600015137000157
38000163390001674000017341000179
42000181430001914400019345000197

```

LAATSTE 10 REGELS 3000 PRIMES:  
PRIMES NR 2951 T/M 3000

```

51026993620270116302701764027031
55027043660270596702706168027067
59027073700270777102709173027103

```

```

73027107740271097502712776027143
77027179780271917902719780027211
81027239820272418302725384027259
85027271860272778702728188027283
89027299900273299102733792027361
93027367940273979502740796027409
97027427980274319902743700027449

```

LAATSTE 98 PRIMES < 1E6:  
PRIMES NR 78401 T/M 78428

```

01998617009986230099862904998633
059986510099865307998658100998657
09998689109986717119986737129986743
139986749149986759159986779169986813
17998681918998683119986839209986843
219986857209986861099868697209986899
2599868917099868927099868941099868947
299986911099868957099868957099868957
3399869303099869007099869023099869029
37998694330998690490998690574099869063
4199869591409986910140998691334099869149
459986969409986918140998691994099869217
4999869821509986923309986923099869259
539986992875099869307099869329099869331
57998699359099869371099869377099869389

```

```


619994316099943309994337099943764999451
6599949160999499994999952100999509
6999954170999553710999563700999599
73999611749996137099962370999631
7799965370999667709996710999683
81999721809997270999749999763
8599976980999773099978099978809997917
8999983390999833099983309998330999833
93999931940999933099993309999330999933
9799997998999983

```

In figuur 4B staan een aantal PRINTOUTS van "per 1" varianten. Wat de laatste betreft, tot mijn verrassing bleken er na de 78400ste PRIEM nog 98 te komen tot 1E6. Ook opmerkelijk vond ik dat het grootste PRIEM-verschil in dat gebied slechts 44 groot is: zie PRIEM 87 en 88. Dit in verband met mijn volgende artikel over het vinden van grote PRIEMWOESTIJNEN.

HAN VAN ABBE

## INFODAG SGGE

 p zaterdag 24 oktober toog de redactie naar Delfzijl om eens te gaan kijken op de Infodag van de Sinclair Gebruikers Groep Eemsmond. De wegbewijzing in Delfzijl was niet helemaal je dat, we moesten even goed zoeken voor we de Molenberg waar dit allemaal georganiseerd was konden vinden. Eenmaal aangekomen vielen we binnen in een overvolle zaal waar computeraars van diverse pluimage aantroffen. Amiga's, Atari's, Spectrum's, QL's, Commodore's, allemaal stonden en zaten ze broederlijk naast elkaar. Er was van alles te zien en te beleven. Er waren diverse demonstraties; we zagen midi-interfaces, PC's, diverse tekstverwerkers, tekenprogramma's, en zelfs een heuse laser. Kortom teveel om op te noemen. Ook waren er diverse firma's naar het hoge noorden gekomen om hun waren (tegen leuke beursprijzen) aan de man te brengen. Zo zagen wij onder andere de firma's Byte, Dataskip, Terminal Software Publicaties en Poking International. Als (hopelijk) de SGGE volgend jaar weer een Infodag organiseert moet je zeker gaan kijken!

De Redactie.

## REKTIFIKATIES

**W**aar gehakt wordt vallen spaanders. Nou, er zijn de vorige maand heel wat spaander gevallen op de redactie. We hebben in elk geval de nodige rektifikaties. Om te beginnen maakten we in het stuk van de redactie dat de firma Eltek (importeur van de Speccies) met de noorderzon was vertrokken. Via via kwamen we erachter dat ze gewoon waren verhuisd van Badhoevedorp naar Barneveld. Dan hadden ze ook maar een adreswijziging moeten sturen. De Noordelijke Computerdagen in Zuidlaren bleken helemaal niet op 26 - 29 oktober te zijn. hoe we aan dit bericht kwamen kunnen we zelf ook niet meer achterhalen. En last but not least zijn er in het artikel "Rekenen met de ZX81 (9)" van Han van Abbe 2 storende foutjes geslopen: op pag. 15 staat bij AD 40F4 MXT, dit moet zijn NXT. Bij 4119 is ? (vraagteken) vergeten. Onze excuses voor deze rektifikaties.

de Redactie.





Op de jaarlijkse **HARD- EN SOFTWAREMARKT** kun je weer spullen meenemen om te verkopen of te ruilen. Veel mensen hebben hier al veel profijt van gehad door spullen voor de computer die ze toch niet meer gebruikten te verkopen of te ruilen voor iets wat ze graag wilden hebben. Ook (originele) software is welkom. Grijp dus je kans en kom op **10** december naar het bekende adres!



# DRUKWERK

Afz.:

SGG  
redactieadres  
Geert Valckeshof 28  
9351 RX LEEK

PORT BETAALD  
ZUIDHORN

AAN:

---

---