COMX-NEWS 34, Jaargang 5 Nummer 3

INHOUDSOPGAVE:

1.....Inhoudsopgave
2.....Van de redactie
3.....Evenredig zwevend
4.....CFORTH voor beginners (1)
5.....CFORTH voor beginners (1)
6.....Wat is AHON?
7.....Wat is AHON?
8.....Do Loop
8.....Hersengymnastiek
9.....Hersengymnastiek
10.....Hersengymnastiek
11.....Clear Screen in CFORTH
12.....CFORTH Tonaal
13.....Faculteitsberekening (2)
14.....Faculteitsberekening (2)
15.....Geen copy
16.....Geen copy
17.....Geen copy
18.....Geen copy

Redactieadres:

Op gen Hoes 113
6442 PR
Brunssum.

Adres C.G.O.N.:
Postbus 7127,
3430 JC,
Nieuwegein.
Giro:
16.71.947

Bank:
33.10.77.787

INZENDEN COPY
AAN: Redactie COMX-NEWS
Op Gen Hoes 113
6442 PR Brunssum
Tel: 045-258454
(19.00-21.00 uur!!)

C.G.O.N. ; Postbus 7127, 3430 JC Nieuwegein
Giro 16.71.947, Bank 33.10.77.787

VAN DE REDACTIE

Voor u ligt COMX NEWS 34.
Het had al nummer 37 kunnen zijn.
U heeft het alvast lang geraden!!! Copy tekort!!
Het voordeel: Voorlopig nog geen contributie betalen.
Het nadeel: Weinig copy geeft weinig nummers.

Mocht ook U als lezer zich afgevraagd hebben waar COMX NEWS bleef, neem dan nu de pen ter hand (of F&M WP) en stuur Uw bijdrage nog deze week in.

Dan nu nog enige opmerkingen:

De listings blijven te bestellen op het C.G.O.N. adres (zie pagina 1 onderaan)

De uitslag van de 6de programmeerwedstrijd en een overzicht van de nieuwe programma's komen in nummer 35.
Met Uw hulp zal dit begin december verschijnen!

Onze oproep in nummer 33 voor een programma-bibliotheek heeft zegge en schrijve 1 reacties opgeleverd.
Mocht U suggesties hebben om ons aller COMX NEWS levend te houden, ???? .... SCHRIJF!!!

Rest mij U nog mede te delen, dat de AHON-programma's ook rechtstreeks bij de makers besteld kunnen worden.
Meer hierover in nummer 35.

Leo Degenkamp

Nagekomen bericht van de importeur:
West Electronics kunt u op de HCC dagen (25/26 november in de Jaarbeurs te Utrecht) vinden op de stand M166.

EVENREDIG ZWEVEND

Ergens in het grijze verleden, toen ons COMX computertje nog maar pas uit was, heb ik in ons clubblad (nr. 5) laten opnemen hoe je met behulp van het TONE commando een chromatische toonladder kunt maken. Dat ging dan als volgt:

```
30 DATA 126,119,112,106,100,94,89,84,79,75,71,67
40 REM  F   F#  G   G#  A   A# B  C  C# D  D# E
```

Hoe kwam ik destijds aan die getallen ?
Volgens de muziektheorie komt een interval van een halve toon overeen met een verhouding in trillingstijden van een twaalfde wortel uit twee. Dit omdat er twaalf halve tonen in een oktaaf passen. We noemen dit de evenredig zwevende stemming.
Met een eenvoudig BASIC programmaatje kan nu de "beste" reeks trillingstijden worden bepaald. Zie de bijgevoegde listing:

```
10 GOTO 140
20 R=0:T=0:D=0
30 FOR K=1 TO 12
40 E=B/2^((K-1)/12):N=FNUM(E)
50 PRINT "TRILLINGSTIJD EXAKT = ";E
60 PRINT "TRILLINGSTIJD AFGEROND = ";N
70 A=ABS(N-E):Q=100*A/E
80 IF Q>D THEN D=Q
90 R=R+A*A:T=T+E*E: NEXT K
100 P=SQR(R/T)*100
110 PRINT "AFWIJKING MAXIMAAL = ";D
120 PRINT "AFWIJKING GLOBAAL = ";P
130 RETURN
140 H=100:G=100:L=0:M=0
150 FOR B=128 TO 64 STEP -1
160 PRINT : PRINT "BASIS = ";B
170 GOSUB 20
180 IF P<G THEN G=P:M=B
190 IF D<H THEN H=D:L=B
200 NEXT B
210 PRINT "BESTE TOONLADDER :"
220 B=M: GOSUB 20
230 PRINT "BESTE TOONLADDER :"
240 B=L: GOSUB 20
250 END
```

Door: Han de Bruijn

CFORTH VOOR BEGINNERS (1)

CFORTH is de COMX versie van de computertaal FORTH en werd in de jaren '70 ontwikkeld door Charles Moore. De taal wordt veel gebruikt in de wetenschap. Enkele voorbeelden: FORTH wordt gebruikt bij het besturen van sterrekijkers. Zelf heb ik al eens gewerkt met een door FORTH bestuurde Fourier Transform Infra Red Spectrofotometer.

Er zijn verschillende FORTH versies bekend, zoals: POLYFORTH, FIG FORTH en FORTH-79. Echter voor alle versies geldt dat ze na enkele aanpassingen overdraagbaar zijn. Dus heb je met deze taal niet meer zoveel last met het omzetten van programma's. Met BASIC zijn we dit wel anders gewend.
De kracht van FORTH schuilt vooral in het feit dat je zelf opdrachten kunt maken, waarmee je een totaal andere taal kunt ontwikkelen. De PASCAL en LOGO, die voor de COMX beschikbaar zijn, zijn ook geheel in FORTH geschreven!
FORTH is uitermate geschikt voor besturing van regelsystemen, omdat je met FORTH vrij gemakkelijk 'n CHIP kunt manipuleren. Omdat de COMX een echte besturingschip in zich heeft, de RCA 1802A, past FORTH uitstekend op ons aller computer.

HOE TE BEGINNEN ?

Voordat we in CFORTH kunnen programmeren zullen we eerst de CFORTH compiler (=vertaler) moeten laden van cassette of floppy. We typen daarna RUN in,er verschijnt dan op het scherm: (computer uitvoer wordt vanaf nu ONDERSTREEPT, dit om het te onderscheiden van wat men zelf intypt.)

COMX CFORTH V5.00

Hieronder staat dan een roze cursor.
Als we nu proberen een BASIC programma in te typen, zult u merken dat de computer reageert met foutmeldingen (Probeer dit uit). We kunnen dus nu niet meer programmeren in BASIC. De BASIC is als het ware vervangen door de FORTH.
Hoe moeten we dan weer terugkomen in de BASIC ? Daarvoor hebben we de beschikking over twee opdrachten. nl. MON en BYE. Voorlopig gebruiken we echter alleen MON (van MONitor). Typ maar eens in:

```
MON            (uiteraard met de CR toets !!)
READY
:_
```

U bent dus nu weer terug in de BASIC. Geef nu weer RUN, en druk dan de CR (Carriage Return) toets in. Er verschijnt nu OK op het scherm. Dit is in CFORTH het teken dat alles goed is gegaan na 'n door u gegeven opdracht. in dit geval de CR-toets indrukken. He equivalent met BASIC is, zoals u wel zult weten, READY.

Om een indruk te krijgen van alle commando's die er zijn in FORTH moet u maar eens VLIST intypen:

VLIST

NONE COL3 COL2 COL1 WHITE MAGENTA .....

U krijgt dus nu een opsomming van alle FORTH opdrachten die er zijn. Als u zelf een opdracht maakt zal deze boven aan de lijst komen te staan. Deze lijst wordt ook wel de FORTH VOCABULARY genoemd. Een voorbeeld van een zelf gemaakte definitie staat hieronder. Typ deze precies zo in, inclusief de spaties. Maakt u zich nog geen zorgen over de werking, dit zal later wel duidelijk worden! (Overbodig misschien: geef na elke regel een carriage return met de CR toets)

```
: TEST          <-- dit typt u in
  HOME CLS ;    <-- dit typt u in
```

Als u nu weer VLIST intypt, dan zult u zien dat TEST bovenaan de lijst staat. (VLIST kunt u stoppen met de ESCape toets) Indien u een typfout hebt gemaakt, dan krijgt u een foutmelding te zien. U begint dan weer gewoon opnieuw. De foutmeldingen komen later aan bod, als we wat verder gevorderd zijn.
Een FORTH programma kunt u RUNNEN, om deze BASIC term maar eens te gebruiken, door eenvoudig de naam van het programma in te typen. In ons voorbeeld hebben we het programmaatje TEST genoemd. Dus typ nu TEST in en let eens op wat er gebeurt.

Schijnbaar zorgt het programma ervoor dat het scherm wordt schoongemaakt vanaf de linkerbovenhoek. We kunnen dus nu de volgende conclusies trekken:

- Elk FORTH programma begint met een dubbele punt (: = COLON)
- Na de dubbele punt volgt dan de naam van het programma, dat men vrij kan kiezen.
- Dan volgen de opdrachten die het programma moet uitvoeren. In ons voorbeeld HOME en CLS.
- Elk programma wordt afgesloten met een punt-komma (;).

Als u kijkt naar de verschillende FORTH programma's die al zijn verschenen in COMX NEWS zult u ontdekken dat alle programma's zo zijn opgebouwd. Dit soort programma's worden ook wel COLON-definities genoemd, vanwege het feit dat ze met een dubbele punt beginnen.
In principe zouden we nu al in FORTH kunnen programmeren. Onze kennis van de verschillende opdrachten is echter nog te gering.

De volgende keer ga ik in op het gebruik van STACK en het rekenen met FORTH.

Reakties zijn altijd welkom op onderstaand adres:

ERWIN SMIT
Beatrixstraat 38
7141 XW Groenlo

WAT IS AHON ?

Voor de COMX-gebruikers die nog niet weten wie of wat AHON is, volgt nu een kleine terugblik in de geschiedenis.
Op 22 december 1983 kocht ik, Oscar Nooy een COMX-35 computer, vanwege de 200 gratis programma's die ik erbij zou krijgen. Toen bleek dat je deze programma's niet geheel gratis kreeg, maar 15 cent per Kilobyte moest betalen en de programma's zelf moest invoeren van listing was ik een beetje gedesillusioneerd. Maar niet getreurd, ik had een vriend die het wel leuk vond om de listings in te voeren, Arjan Houben. Zo maakte hij ook kennis met de COMX, maar had geen interesse om er een te kopen, vooral omdat er nog niet veel goede spellen verkrijgbaar waren. In 1985 ruilde ik mijn COMX-35 in voor een COMX PC-1 met bijbetaling van f 298,-. Voor Kerstmis van dat jaar kocht Arjan een tweedehandse COMX-35 die hij bij de volgende inruilactie omruilde voor een PC-1. We noemden ons toen al AHON, naar onze initialen Arjan Houben en Oscar Nooy.
We vonden dat er meer uit de COMX te halen viel dan er tot op dat moment gebeurde. Zodoende begonnen we onze eerste programma's te maken, in BASIC natuurlijk. Nederland en Fun factory stammen uit die tijd. Daarna begon ik machinetaal te leren. Met deze nieuwe programeertaal gewapend begonnen we aan een project waarvan we de uitwerking nooit hadden verwacht : HIT IT waarmee we in de zesde programmeerwedstrijd COMX spullen twv DUIZEND gulden verdienden, zodat we in totaal spullen (TV, 4*COMX-35, th.printer enz.) kregen twv f 1700,-.
Inmiddels had ik Arjan ook de eerste machinetaal beginselen bijgebracht, en door veel oefenen werd hij de taal ook machtig. Hij begon toen met Killerwatt, dat we daarna samen afmaakten. Vervolgens hebben we met ons beider capaciteiten de spellen Flying Carpet en Boulderdash gemaakt. Deze drie spellen nemen elk ca. 30K van het geheugen in beslag, en zijn elk in machinetaal geschreven. Ze bevatten zo'n 25 velden, en Boulderdash zelfs TACHTIG ! Je zou bijna denken dat deze programma's op elkaar lijken, maar het tegendeel is waar. Vraag maar aan de mensen die naar de laatste paar COMX-gebruikersdagen zijn geweest. Voor het maken van Killerwatt en Flying Carpet hebben we in totaal ruim drie weken (niet achter elkaar) gewerkt. Deze drie super-spellen hebben we al ingezonden voor de 7de programmeerwedstrijd, en we hopen op een goede uitslag.

Na een examenreces van mij (met goed gevolg), zijn we weer druk bezig met HERO, weer zo'n lang programma, en Streetracer wat weer een iets korter programma wordt. Het programma AHON-spellen zal waarschijnlijk niet veel verder worden uitgebreid, omdat ik in Eindhoven ga studeren, en op kamers ga. Het zal hierdoor moeilijk worden om nog veel te programmeren. Misschien dat Arjan nog doorgaat, en zijn programma's onder AHON zal uitbrengen.

Het beste van Ahon vinden wij de graphics en het slechtste de muziek, daar hebben wij helemaal geen kaas van gegeten. Ondanks dat vinden wij, zonder eigendunk, dat onze spellen zeer geslaagd zijn.

U moet nu niet denken dat U niets meer van AHON zal horen, de meesten van U zullen pas binnenkort met de betere spellen van AHON kennismaken, en wel Killerwatt, Flying Carpet, Boulderdash, HERO en Streetracer.(samen bijna 150Kb lang) Wij hopen U hiermee veel speelplezier te bieden.

Onze uitrusting is op het moment als volgt :

Oscar

1 COMX PC-1
1 kleurenmonitor
1 z/w tv
2 COMX-35
2 Datarecorders
1 80 koloms thermoprinter
1 COMX-35p thermoprinter
2 Dubbelzijdige diskdrives
2 Joysticks
1 Floppy Disk Controller
1 Serial Card
1 STP Printerkaart
1 Thermal PTR kaart
1 Joykaart
1 Expansion box
92 Dubbelzijdige diskettes

Arjan

1 COMX PC-1
1 z/w/tv
2 COMX-35
1 Datarecorder
1 COMX-35p thermoprinter
1 enkelzijdige diskdrive
2 Joysticks
1 Floppy Disk controller
1 Serial card
1 Thermal ptr kaart
1 Joykaart
1 expansion box
30 Dubbele enkelzijdige diskettes

Moraal van het verhaal :
Met 1 COMX-35, een datarecorder en zonder computerkennis kan een persoon uitgroeien tot een tweemans software productieteam, dat als een van de betere COMX programmers beschouwd mag worden.
run1000

Geschreven door Oscar Nooy en Arjan Houben, ofwel AHON

DO,.... LOOP, IN DE FORTH ASSEMBLER

Bij een revisie van de FORTH ASSEMBLER stuitte ik onlangs op de definities van DO en LOOP, (in de assembler). Met de definitie op zich is niets aan de hand, het werkt. Zoals met veel definities echter het geval is: het kan netter.

```
ASSEMBLER DEFINITIONS
: 0>=WHILE,  0>=IF, ;
: DO, SWAP OVER LDR,
  BEGIN,    ,
        SWAP DUP DEC, GHI,
             0>=WHILE, ;
: LOOP, REPEAT, ;
```

In de oude definitie was de aanroep:
..N DO, ...MNEMONIC,s ... LOOP, ...
In de nieuwe definitie wordt de aanroep:
..N REG DO, ...MNEMONIC,s ... LOOP, ...
N is hierin een 16 bits getal.
REG is hierin het register nummer (LOOPcounter).

Door: Yuen Keong Ng

HERSENGYMNASTIEK

De puzzel is om twee getallen te vinden.
Gegeven is dat ze allebei groter zijn dan 1 en kleiner dan 100.
Verder zijn er twee personen in het spel. De een, Simon (S), weet alleen de som van de getallen. De ander, Pieter (P), kent alleen het produkt van de getallen.

Simon en Pieter hebben met elkaar het volgende gesprek:
P: Ik weet niet wat de getallen zijn.
S: Dat wist ik al ! Maar ik weet ze ook niet.
P : O, maar dan weet ik ze nu !
S : O, maar dan weet ik ze nu ook !

Kunt u, beste lezers, op grond van deze informatie achterhalen over welke twee getallen het gaat ? Uw COMX kan bij het oplossen van deze puzzel goede diensten bewijzen !

```
10 GOTO 120
20 GOSUB 30: PRINT D: END: REM *TEST*
30 REM ** BEPALEN VAN DELERS **
40 D=0: FOR A=2 TO 100
50 IF A*A>P THEN  EXIT 100
60 IF MOD(P,A)<>0 THEN  GOTO 90
70 B=P/A: IF B>100 THEN  GOTO 90
80 D=D+1
90 NEXT A
100 RETURN
110 REM P: IK KEN DE GETALLEN NIET
120 REM S: DAT WIST IK. IK OOK NIET.
130 DEFINT Z: DIM S(200)
140 N=0: FOR T=4 TO 200
150 N=N+1:S(N)=T
160 FOR X=2 TO T/2
170 Y=T-X: IF Y>100 THEN  GOTO 210
180 P=X*Y: GOSUB 30
190 IF D>1 THEN  GOTO 210
200 N=N-1: EXIT 220
210 NEXT X
220 PRINT "S=";T;" N=";N: NEXT T
230 CALL (@C060): REM PRINTER AAN
240 PRINT : PRINT "AANTAL = ";N
250 PRINT "GELDIGE SOMMEN:"
260 FOR K=1 TO N
270 PRINT S(K);" ";: NEXT K
280 PRINT : CALL (@C050): GOTO 360
290 REM ** EERSTE DEEL OVERSLAAN **
300 RESTORE: DEFINT Z
310 DIM S(10):N=10: FOR K=1 TO N
320 READ S(K): NEXT K
330 REM HIER STONDEN BASIC "DATA" !!
340 GOTO 230
350 REM P: O, MAAR DAN WEET IK HET !
360 M=0: FOR K=1 TO N
370 M=M+(S(K)-1)/2-1: NEXT K
380 DIM G(M),H(M):M=0
390 FOR I=1 TO N-1:T=S(I)
400 FOR X=2 TO T/2:Y=T-X:P=X*Y
410 M=M+1:G(M)=T:H(M)=P
420 FOR J=1 TO N: IF I=J THEN  GOTO 490
430 C=S(J): FOR A=2 TO C/2:B=C-A:Q=A*B
440 IF P<>Q THEN  GOTO 460
450 EXIT 480
460 NEXT A
470 GOTO 490
480 M=M-1: EXIT 500
490 NEXT J
500 PRINT M,T,P: NEXT X: NEXT I
505 CALL (@C060): REM PRINTER AAN
510 PRINT : PRINT "GELDIGE PRODUKTEN:"
```



```
520 N=0:I=0
530 FOR K=1 TO M:S=G(K):P=H(K)
540 IF S=N THEN  GOTO 570
550 PRINT : PRINT : PRINT "SOM = ";S;
560 N=S:I=0
570 L=MOD(I,10): IF L=0 THEN  PRINT
580 PRINT TAB(4*L);P;
590 I=I+1: NEXT K: CALL (@C050): END
```

Door: Han de Bruijn

Hier
Had
Uw
Bijdrage
Kunnen
Staan

CLEAR SCREEN (CLS) IN CFORTH

Een van de problemen bij het realiseren van CFORTH ROM was de CLS routine. De aanwezige routine maakte gebruik van adressen in het BASIC ROM geheugen. De oude routine zag er als volgt uit:

```
: CLS
  FBBF F800 DO 0 I OUTSCR LOOP ;
```

OUTSCR is een machinetaal woord. Dit woord gebruikt enkele adressen van de BASIC. In een CFORTH ROM kan dit natuurlijk niet. Daarom heb ik een versie van CLS geschreven waarbij dit probleem is verholpen. Dit programma of beter definitie staat hieronder. Typ het in met de editor of rechtstreeks.

```
( CLS IN MACHINECODE )
HEX
CREATE CLS
   F8F7 , BE C, F8FF , AE C,
   HERE DUP 1E C, F800 ,
   HERE 34 C, C,
   5E C, 8E C, FBBF , CA C, ,
   9E C, FBFB , CA C, ,
   DC C,
LATEST SMUDGE
DECIMAL
```

Met deze definitie wordt het HELE scherm schoongemaakt. Gedeeltes wissen kan dus hier niet mee.
Voor de machinetaal/FORTH freaks: als je het adres in register E laat afhangen van de CPOS waarde, dan is gedeeltes wissen ook geen probleem meer. Bijvoorbeeld:

```
2 0 CPOS CLS
```

dan moet het beginadres @F800 + 2 * #28 - 1 zijn. Waarbij de 2 staat voor de regel vanaf waar gewist moet worden.

Door: Erwin Smit.

CFORTH TONAAL

Aantekeningen over mijn COMX hobby worden doorgaans opgeborgen in twee grote mappen: een voor muziek, en een voor de rest. Het is een wonder hoe ik al die tijd een artikeltje over muziek heb kunnen uitstellen, maar nu komt het er dan eindelijk van.

Om niet al te rigoureus met het verleden te breken, treft u aan een van BASIC ROM onafhankelijke TOON routine:

```
HEX CREATE TOON  1909 , FF01 , BE C,  ( frequentie in RE.1 )
2929 , 09 C, FF01 , FEFE , FEFE , ( oktaaf links in accu D )
2929 , E9F1 , AE C, ( volume rechts in D, samen naar RE.0 )
EE64 , 2929 , 29DC , ( output geluid en netjes terugkeren )
SMUDGE DECIMAL .
```

Of deze TOON routine inderdaad precies hetzelfde presteert als de standaard meegeleverde TONE funktie, kunt u zelf kontroleren.

Maar als je een TOON hebt, heb je daarmee nog geen MUZIEK, of in gewoon BASIC/CFORTH engels: MUSIC. Ik heb hier voor me liggen een verhaal over de frequentie-verhoudingen die voorkomen in de zogenaamde natuurlijke (majeur) toonladder. Gemeten ten opzichte van een "centrale C" liggen die verhoudingen als volgt:

C	D	E	F	G	A	B	C
1	10/9	5/4	4/3	3/2	5/3	15/8	2

Het kleinste gemene veelvoud van de tellers in deze reeks is 60. We kunnen dus ook schrijven:

C	D	E	F	G	A	B	C
1/60	1/54	1/48	1/45	1/40	1/36	1/32	1/30

Waarom doen we dat ? In tegenstelling tot wat iedereen schijnt te denken wordt als invoer voor de TONE funktie namelijk niet een frequentie opgegeven, maar het omgekeerde daarvan: de z.g. trillingstijd. Weten we dat nog uit de natuurkunde les ? Bovenstaande getallen herkennen we dan ook in de volgende FORTH definitie:

```
60 VARIABLE REIN 54 , 48 , 45 , 40 , 36 , 32 ,
```

Door: Han de Bruijn

FACULTEITSBEREKENING (2)

Het is altijd weer aardig om te zien hoe een computer iets berekend wat voor mensen te moeilijk is of dat anders te lang zou duren.
In COMX NEWS 19 had ik een stukje geschreven over hoe men de faculteit kan berekenen in 3 verschillende talen. Nadeel van de de gegeven programma's was dat men gebonden was aan een bepaalde maximale waarde. Bij PASCAL en CFORTH was dat 7! en bij BASIC 33!. De oorzaak hiervan is dat bij het berekenen van de faculteit de uitkomsten snel heel groot worden. De computer kan deze getallen dan niet meer verwerken. (Bij CFORTH kan men de grens nog iets verleggen door gebruik te maken van dubbele precisie getallen.)
Hieronder geef ik een programma in BASIC en in CFORTH dat aan bovengenoemd probleem een einde maakt. Alleerst de programma's en dan de uitleg.

BASIC

```
10 REM FACULTEITEN
20 DEFINT Z
30 DIM B(255): FOR I=1 TO 255: B(I)=0: NEXT : B(255)=1
40 CPOS(0,0):CLS
50 FOR G=2 TO 255
60 PRINT: PRINT G-1;"!=";
70 A=0
80 FOR I=1 TO 255
90 IF A+B(I)=0 GOTO 130
100 A=1
110 PRINT MOD(B(I),10);
120 B(I)=G*B(I)
130 NEXT
140 FOR I=255 TO 1 STEP -1
150 IF B(I)<10 GOTO 190
160 A=INT(B(I)/10)
170 B(I-1)=B(I-1)+A
180 B(I)=B(I)-10*A
190 NEXT:NEXT
```

Dit programma kan een faculteit berekenen met een uitkomst van maximaal 255 getallen !! De uitkomst wordt ook als zodanig uitgeprint. Het enige nadeel is dat als je de faculteit van 65 wilt hebben, je 24 minuten moet wachten. Maar daar staat tegenover dat je de uitkomst in alle 99 cijfers nauwkeurig hebt!
De uitkomst wordt precies hetzelfde uitgerekend als een vermenigvuldiging met de hand. De uitkomst wordt gezet in ARRAY B, met de eenheden in B(255), de tientallen in B(254) enz.

CFORTH

Hier wordt de faculteit op dezelfde manier uitgerekend als bij het basic programma. Het voordeel van FORTH is de snelheid en dat men niet gebonden is aan de array grootte. Men kan dus de uitkomst weergeven in duizenden, ja zelfs een miljoen getallen.

Men kan dit zelf bepalen door een grotere array te maken. Enige beperking is de grootte van 't beschikbare geheugen. Ik heb hier gekozen voor een maximale grootte van 255 getallen, zodat de twee programma's konden worden vergeleken.
Het CFORTH programma heeft 12 minuten nodig om 65! te berekenen. Dit is dus een aanzienlijke tijdwinst.
In het programma heb ik dezelfde variabelen gebruikt als in het BASIC programma, zodat men ze kan vergelijken.

```
0 VARIABLE HOEVAAK 0 VARIABLE A 0 VARIABLE G
: ARRAY <BUILDS 2* ALLOT DOES> SWAP 2* + ;
270 ARRAY B
: EMPTY  256 1 DO 0 I B ! LOOP ;
: FAC
  2+ HOEVAAK ! HOME CLS ." FACULTEITEN" CR CR
  EMPTY 1 255 B !
  HOEVAAK @ 2 DO I G !
  CR ." DE FACULTEIT VAN " I 1- . ." IS" CR CR
  0 A !
  256 1 DO I B @ A @ + 0 <> IF
  ?TERMINAL IF ."   OK" CR QUIT THEN
  1 A ! I B @ 10 MOD . ." cntl-j"
  G @ I B @ * I B ! THEN LOOP
  0 255 DO I B @ 10 >= IF
  I B @ 10 / A ! I 1- B @ A @ + I 1- B !
  I B @ 10 A @ * - I B !
  THEN -1 +LOOP CR LOOP ;
```

Dit programma kan men onderbreken met ESCape. Om het programma te starten, kan men het volgende intypen: 100 FAC waarbij 100 het laatste te bepalen getal is.

Om te controleren of de programma's goed werken geef ik hieronder de eerste 13 oplossingen.

```
 1!= 1            2!= 2            3!= 6
 4!= 24           5!= 120          6!= 720
 7!= 5040         8!= 40320        9!= 362880
10!= 3628800     11!= 39916800    12!= 479001600
13!= 6227020800
```

Door: Erwin Smit