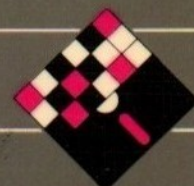


PROFESSIONELE SOFTWARE VOOR DE

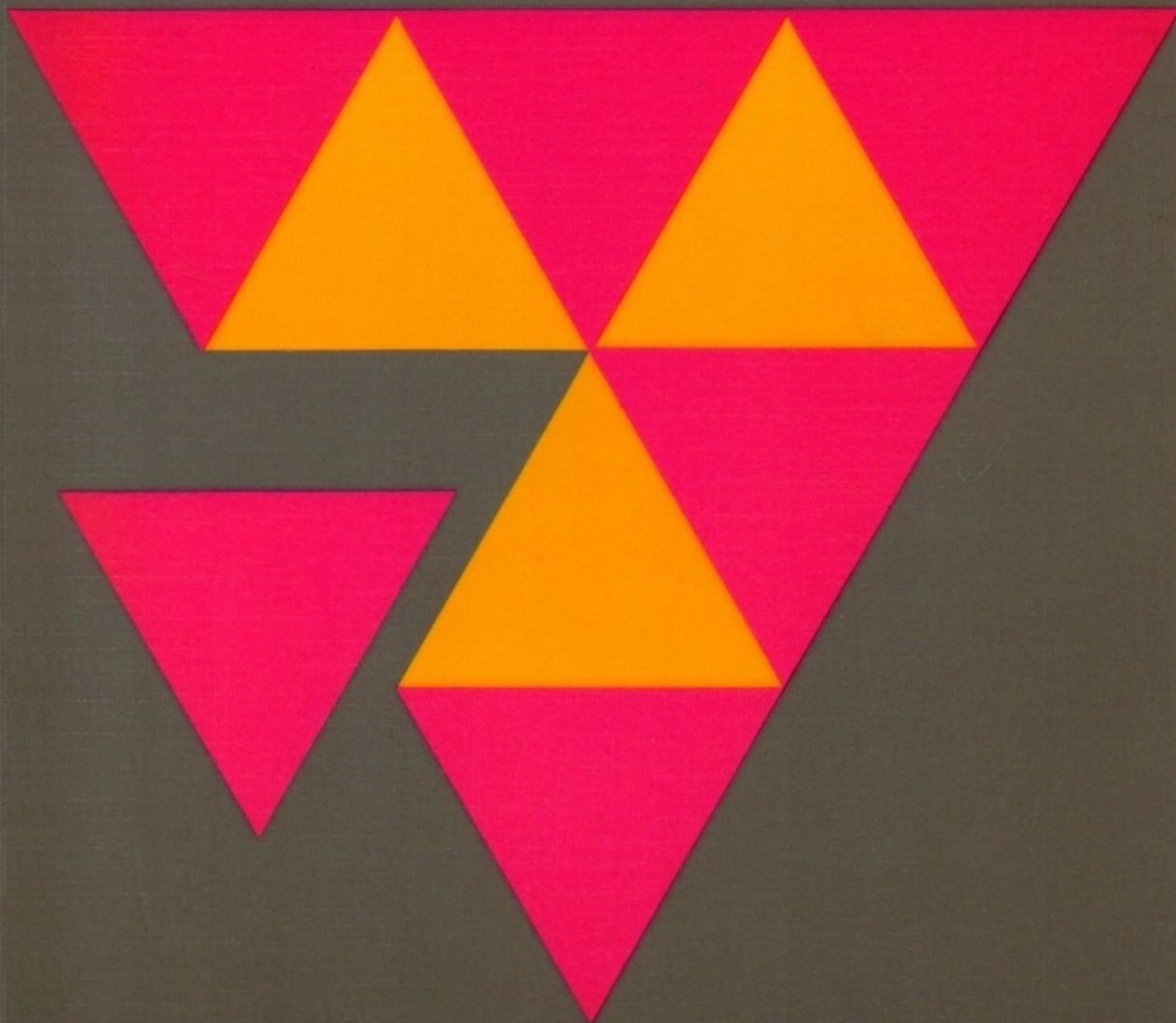
# MSX COMPUTER

voor MSX1 en MSX 2



PROFESSIONELE  
SOFTWARE REEKS

DR. IR. K.L. BOON / IR. A.R.TH. PELSMAEKER



BOON/PELSMAEKER  
PROFESSIONELE SOFTWARE VOOR DE MSX COMPUTER





**MSX1/MSX2**



**Professionele Software Reeks**

# **MSX1/MSX2**

Software voor  
Zakelijke en Technische  
Toepassingen

Onder redactie van:

Dr Ir K.L. Boon  
Ir A.R.Th. Pelsmaeker



**Addison-Wesley Europe B.V.**



**Uitgeverij OMIKRON**

Omslagontwerp: BSE Advertising - Alphen a/d Rijn

Met dank aan:

H.J. Boonstra  
R. Hermsen  
C. Baten  
H.B.A. Groot Lipman  
J.E.F. Groot Lipman  
A. van Denzel  
C.B. Dispa

## CIP-GEGEVENS

MSX

MSX computer : software voor zakelijke en technische toepassingen  
onder red. van K.L. Boon, A.R.Th. Pelsmaeker. -  
Amsterdam : Addison-Wesley Europe ; Enschede : Omikron.  
(Professionele software reeks)  
ISBN 90-6789-051-0  
SISO 365.2 UDC 681.3.06  
Trefw.: software / MSX (computer).

1e druk 1986

© 1986 Uitgeverij Omikron, Enschede

Niets uit deze uitgave mag worden vermenigvuldigd of openbaar worden gemaakt door middel van druk, fotokopie of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van Uitgeverij Omikron.

Ondanks alle aan de samenstelling van dit boek bestede zorg, kan noch de redactie noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade, die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voorkomen.

# VOORWOORD

Voor u ligt een verzameling programma's die direct toepasbaar zijn op de MSX computer. Onder de vele boeken die reeds over deze computer gepubliceerd zijn, zal dit boek waarschijnlijk een bijzondere plaats innemen en wel door de veelheid van programma's op zoveel verschillende gebieden. Dit boek toont dan ook duidelijk aan dat de MSX een zeer veelzijdige computer is en zich uitstekend leent voor allerlei technische en zakelijke toepassingen.

Behalve algemene wiskundige en statistische toepassingen zijn onder andere simulatie-programma's, een grafisch-programma, een tekstverwerkingsprogramma en een databestandenprogramma opgenomen. Bij alle programma's is een voorbeeld geplaatst, zodat u eenvoudig kunt controleren of alles goed is ingetoetst.

Bij de grote programma's is bovendien gebruik gemaakt van checksums, zodat de controle zeer nauwkeurig is uit te voeren.

Tot slot nog een speciaal woord van dank aan Harm-Jan Boonstra voor de realisatie van vele van deze programma's waaronder de schaakklok en het muziekprogramma „songgenerator“.

De Samenstellers

# INHOUD

## 1. WISKUNDIGE PROGRAMMA'S

1.1.1	Priemgetallen	9
1.1.2	Ontbinden in factoren	11
1.2	Het omzetten van getallen	12
1.3	Reeksen	15
1.3.1	Reeks van Fibonacci	15
1.3.2	Bernoulli getallen	17
1.4	Complexe getallen	20
1.4.1	Complexe getallen 1	20
1.4.2	Complexe getallen: bewerkingen	23
1.5	E-machten	27
1.6	Faculteiten	28
1.6.1	Stirling-benadering	28
1.6.2	Faculteit	29
1.7	Integreren en differentiëren	30
1.7.1	Integreren	30
1.7.2	Differentiëren	32
1.8	Matrixbewerkingen	34
1.8.1	Vereenvoudigen van matrices	34
1.8.2	Het bepalen van de determinant	38
1.8.3	Het oplossen van een stelsel lineaire vergelijkingen	41
1.8.4	Inverse matrix	46
1.9	De errorfunctie	49
1.10	De Fouriertransformatie en het schatten van spectra	52
1.11	Vektorbewerkingen	57
1.11.1	Lengte van een vector	58
1.11.2	Hoek tussen twee vectoren	59
1.12	Nulpuntsbepaling	62

## 2. STATISTISCHE PROGRAMMA'S

2.1	Kansverdelingen	64
2.1.1	Binomiaalverdeling	64
2.1.2	Poissonverdeling	66
2.1.3	De standaardnormaalverdeling	68
2.2	Histogram	70
2.3	Gemiddelde en standaarddeviatie	73
2.4	Covariantie en correlatie-coëfficiënt	75
2.5	Curvefitting	78
2.6	Combinaties en permutaties	82

## 3. FINANCIËLE BEREKENINGEN

3.1	Bepalen van totaal gespaard bedrag	85
3.2	Bepalen van periodiek bedrag	87



3.3	Klimsparen	89
3.4	Periodieke uitkering	90
3.5	Persoonlijke lening	92
3.6	Annuïteitenhypotheek	93
3.7	Lineaire hypotheek	95
3.8	Rentabiliteit	98

#### 4. ALGEMEEN

4.1.1	Verzwakkers	102
4.1.2	Serieel netwerk	105
4.1.3	Parallel netwerk	109
4.2	Songgenerator	113
4.3	Rembrandts toolkit	127
4.4	Schaakklok	134
4.5	Bugger, een programmeerhulp	137
4.6	Checksumgenerator	142

#### 5. SIMULATIE

5.1	Inleiding	146
5.2	Analoge simulatie	146
5.2.1	De bouwstenen	146
5.2.2	Een uitgewerkt voorbeeld	150
5.3	Digitale simulatie	167
5.3.1	De bouwstenen	167
5.3.2	Een uitgewerkt voorbeeld	167
5.3.3	Invoeren van een model	169
5.3.4	De commando's	170

#### 6. SPELLETJES

6.1	Het letterspel	189
6.2	Mini Mancala	191
6.3	ABBA-spel	195
6.4	Code kraken	198
6.5	Explosiespel	200
6.6	Do you mind, Master	206

#### 7. TEKSTVERWERKER EN DATABESTANDEN

7.1	Tekstverwerker	208
7.2	Databestanden	222
7.3	Mail/Merge	241

8. ASSEMBLER/DISASSEMBLER	246
---------------------------	-----



# 1 WISKUNDIGE PROGRAMMA'S

In dit hoofdstuk zullen we ons bezighouden met allerlei wiskundige toepassingen. Tenslotte is onze MSX een computer, dat wil zeggen een machine waarmee gerekend kan worden. Het aantal wiskundige toepassingen is in feite ongekend groot en daarom hebben we slechts een keuze kunnen maken. Bij die keuze hebben we ons laten leiden door de gebruikelijke verzamelingen BASIC-subroutines die voor wiskundige toepassingen gegeven worden.

We starten met enkele bekende onderwerpen uit de getallentheorie, hierna staan we stil bij het onderwerp „het omzetten van getallen”. Dit gedeelte zal vooral voor programmeurs die in assembler willen werken van belang zijn; assembler-programmeren houdt immers het werken met binaire en hexadecimale getallen in.

Vervolgens komen de complexe getallen en enkele belangrijke functies aan de orde. Uiteraard verschaffen we programma's voor differentiaal- en integraalberekeningen, programma's voor het werken met matrices, het bepalen van de error-functie en het bepalen van de Fouriertransformatie. Eveneens zijn een tweetal vektorprogramma's opgenomen en een programma dat het nulpunt van een functie berekent.

## 1.1.1 PRIEMGETALLEN

Priemgetallen zijn gehele getallen die bij deling alleen een geheel getal opleveren als we door het getal zelf of door 1 delen. Zo is 5 bijvoorbeeld een priemgetal, dit getal is immers alleen deelbaar door 1 en 5. Priemgetallen zijn voor wiskundigen van formaat steeds een geweldige uitdaging geweest. Met name de vraag of men een zeker verband tussen de opeenvolgende getallen zou kunnen ontdekken heeft steeds velen geïntrigeerd. In het gezaghebbende blad Scientific American treft men hierover met de regelmaat van de klok (of met de regelmaat van „priemgetallen”.....) artikelen aan.

Het volgende programma berekent alle priemgetallen tussen twee opgegeven grenzen. Het aantal waarden dat men kan verkrijgen zal uiteindelijk door de beschikbare geheugenruimte worden bepaald.

Voorbeeld:

```
**** priemgetallen ****  
  
beginwaarde? 100  
eindwaarde? 200  
  
tussenliggende priemgetallen  
  
101 103 107 109 113 127 131 137  
139 149 151 157 163 167 173  
179 181 191 193 197 199
```

Programma:

```
10 REM *** priemgetallen ***  
20 REM ** input **  
30 CLS:LOCATE 6,3  
40 PRINT"**** priemgetallen ****"  
50 PRINT:INPUT"beginwaarde";X  
60 INPUT" eindwaarde";Y  
70 DIM PR(Y-X)  
80 PRINT:PRINT"tussenliggende priemgetallen":PRIN  
T  
90 GOSUB 150  
100 FOR A=0 TO Y-X  
110 IF PR(A)=0 THEN PRINT (A+X);  
120 NEXT A  
130 PRINT:PRINT  
140 END  
150 REM ** berekening **  
160 X1=X  
170 P=P+1-(NOT(P=2))  
180 IF P>SQR(Y) THEN RETURN  
190 IF X1/P-INT(X1/P)=0 THEN GOSUB 220:GOTO 160  
200 X1=X1+1:GOTO 190  
210 REM * wegstrepen niet-priemgetallen *  
220 IF X1>Y THEN RETURN  
230 IF X1<>P THEN PR(X1-X)=1  
240 X1=X1+P  
250 GOTO 220
```

## 1.1.2 ONTBINDEN IN FACTOREN

Het onderwerp „ontbinden in factoren” is uiteraard nauw verwant aan het onderwerp „priemgetallen”. Het gaat er hierbij immers om een getal als een vermenigvuldiging van priemgetallen voor te stellen, bijvoorbeeld:  $16 = 2 * 2 * 2 * 2$ .

Als een getal in het geheel niet in factoren te ontbinden is ...jawel, omdat het een priemgetal is, dan zal dit programma dat ook melden.

*Voorbeeld 1:*

```
****   ontbinden in factoren   ****
****           en             ****
****  bepalen van priemgetallen ****

te ontbinden getal ? 510510

510510 = 2 * 3 * 5 * 7 * 11 * 13 * 17
```

*Voorbeeld 2:*

```
****   ontbinden in factoren   ****
****           en             ****
****  bepalen van priemgetallen ****

te ontbinden getal ? 109

109 is een priemgetal
```

*Programma:*

```
10 REM *** ontbinden in factoren ***
20 REM *** of het bepalen van      ***
30 REM *** priemgetallen          ***
40 REM ** invoer van getal **
50 CLS
60 GOSUB 100
70 REM ** berekeningen en uitvoer **
```

```

80 GOSUB 190
90 END
100 REM ** invoer van het getal **
110 LOCATE 1,2
120 PRINT"****   ontbinden in factoren   ****"
130 PRINT" ****           en           ****"
140 PRINT" ****  bepalen van priemgetallen  ****"
150 LOCATE 0,8
160 INPUT" te ontbinden getal ";B
170 PRINT
180 RETURN
190 REM ** berekeningen en uitvoer **
200 PRINT:PRINT STR$(B);
210 A=B:P=2
220 MX=INT(SQR(A))
230 IF P>MX THEN 340
240 IF A=1 THEN 370
250 F=A/P
260 IF F<>INT(F) THEN 310
270 IF A=B THEN PRINT" =";STR$(P);:GOTO 290
280 PRINT" *";STR$(P);
290 A=F
300 GOTO 240
310 IF P=2 THEN P=1
320 P=P+2
330 GOTO 230
340 IF A=B THEN PRINT" is een priemgetal":GOTO 37
0
350 IF A=1 THEN 370
360 PRINT" *";STR$(A);
370 PRINT:PRINT
380 RETURN

```

## 1.2 HET OMZETTEN VAN GETALLEN

Dit programma heeft betrekking op het omzetten van getallen in een ander talstelsel. Het zal vooral van belang zijn voor programmeurs die in assembler willen werken. Bij assembler spelen de zogenaamde binaire en hexadecimale getallen een grote rol. Alvorens we hier dieper op ingaan bespreken we eerst de opbouw van de ons zo vertrouwde (decimale) getallen.

Bij onze decimale getallen gebruiken we 10 symbolen (0 - 9). De plaats van zo'n symbool in een getal correspondeert dan met een macht van 10.

We illustreren dit aan de hand van een voorbeeld.

Het getal 478 is opgebouwd uit 8 eenheden (10 tot de macht 0), 7 tientallen (10 tot de macht 1) en 4 honderdtallen (10 tot de macht 2):



Voorbeeld 2:

```
      **** conversie ****

te converteren getal? 13D
grondtal hiervan? 16
grondtal v/h nieuwe stelsel? 10

in het 10 tallig stelsel is het
bovenstaande getal gelijk aan:

317
```

Programma:

```
10 REM *** conversie ***
20 CLS:LOCATE 7,2
30 PRINT"**** conversie ****"
40 REM ** invoer van gegevens **
50 GOSUB 140
60 REM ** converteer opgegeven getal **
70 REM ** naar dec. representatie **
80 GOSUB 200
90 REM ** conversie **
100 GOSUB 300
110 REM ** uitvoer van gegevens **
120 GOSUB 400
130 END
140 REM ** invoer van gegevens **
150 LOCATE 0,5
160 INPUT"te converteren getal";GT$
170 PRINT:INPUT"grondtal hiervan";G1
180 PRINT:INPUT"grondtal v/h nieuwe stelsel";G2
190 RETURN
200 REM ** bepalen van dec. repr. van gt1$ **
210 GT=0
220 L=LEN(GT$)
230 FOR I=1 TO L
240 CF$=MID$(GT$,I,1)
250 IF CF$<="9" THEN CF=VAL(CF$)
260 IF CF$>="A" AND CF$<="Z" THEN CF=ASC(CF$)-55
```



```

265 IF CF$>="a" AND CF$<="z" THEN CF=ASC(CF$)-87
267 IF CF>=G1 THEN PRINT:PRINT" Invoer past niet
bij grondtal !!":END
270 GT=GT*G1+CF
280 NEXT I
290 RETURN
300 REM ** conversie **
310 RT$=""
320 IF GT=0 THEN RETURN
330 HULP=GT/G2
340 GT=INT(HULP)
350 HULP=(HULP-GT)*G2
360 HULP=INT(HULP+.1)' * afronden *
370 IF HULP<=9 THEN RT$=RT$+CHR$(HULP+48)
380 IF HULP>9 THEN RT$=RT$+CHR$(HULP+55)
390 GOTO 320
400 REM * uitvoer van resultaat *
410 L=LEN(RT$)
420 CF$=""
430 FOR I=0 TO L-1
440 CF$=CF$+MID$(RT$,L-I,1)
450 NEXT I
460 LOCATE 0,12
470 PRINT"in het";G2;"tallig ";
480 PRINT"stelsel is het"
490 PRINT:PRINT"bovenstaande getal gelijk aan:"
500 PRINT:PRINT CF$:PRINT:PRINT
510 RETURN

```

### 1.3 REEKSEN

Een reeks is een serie getallen waarbij een gegeven verband tussen de opeenvolgende waarden aanwezig is. Van de schoolbanken kennen we nog de rekenkundige reeks (iedere volgende term toont een constant verschil met de vorige term) en de meetkundige reeks (iedere volgende term volgt uit vermenigvuldiging met een constante). De nu volgende programma's geven de reeks van Fibonacci en de Bernoulli getallen.

#### 1.3.1 REEKS VAN FIBONACCI

Fibonacci, bijgenaamd Leonardo van Pisa, reisde als koopman van Italië naar verschillende Arabische landen. In 1202 schreef hij een groot handboek over het rekenen met het Hindoe-Arabische getallensysteem. In 1220 beschrijft hij de aanwezige kennis op het gebied van meetkunde en trigonometrie.

We geven deze informatie om duidelijk te maken dat Fibonacci meer is dan een Italiaanse wiskundige die een „aardige” naar hem genoemde reeks bedacht. De lezer die meer informatie wenst verwijzen we naar het klassieke werk van onze landgenoot D.J. Struik, „Geschiedenis van de wiskunde” (Aula, 195).

Nu dan de reeks van Fibonacci. Aan deze reeks ligt het vraagstuk ten grondslag hoeveel konijnen uit één paar kunnen voortkomen, bijvoorbeeld in één jaar, als er elke maand een nieuwe generatie ontstaat die zich vanaf de tweede maand begint voort te planten en als geen enkel konijn in deze periode sterft.

De beoogde reeks heeft de volgende gedaante:

1 1 2 3 5 8

met andere woorden: elke term is de som van de twee voorgaande termen. Deze reeks heeft, behalve dat ze geënt is op een aan de natuur ontleend vraagstuk, enkele markante eigenschappen. Een van die eigenschappen is dat het quotiënt van twee opeenvolgende termen steeds meer naar een constante waarde tendeert, deze waarde komt dan precies overeen met de bekende „gulden snede”.

*Voorbeeld:*

```
**** Fibonacci ****  
  
eerste getal? 1  
tweede getal? 1  
  
te berekenen term? 13  
  
de 13 de term is: 233  
de 14 de term is: 377  
de 15 de term is: 610  
de 16 de term is: 987  
volgende term? (j/n)  
Ok
```

*Programma:*

```
10 REM *** Fibonacci ***  
20 KEY OFF:CLS:LOCATE 7,2  
30 PRINT"**** Fibonacci ****"  
40 REM ** invoer gegevens **  
50 GOSUB 140  
60 REM ** berekening **  
70 GOSUB 220  
80 REM ** uitvoer resultaat **  
90 GOSUB 300
```

```

100 REM ** volgende term **
110 GOSUB 330
120 GOTO 60
130 END
140 REM ** invoer gegevens **
150 LOCATE 0,4
160 INPUT" eerste getal";F1
170 INPUT" tweede getal";F2
180 PRINT:INPUT"te berekenen term";N
190 PRINT
200 B=3
210 RETURN
220 REM ** berekening **
230 IF N<3 THEN 290
240 FOR I=B TO N
250 F3=F1+F2
260 F1=F2
270 F2=F3
280 NEXT I
290 RETURN
300 REM ** uitvoer resultaat **
310 PRINT:PRINT"de";N;"de term is:";CSNG(F2)
320 RETURN
330 REM ** volgende term **
340 PRINT" volgende term? (j/n)
350 A$=INKEY$:IF A$="" THEN 350
360 IF A$="N" OR A$="n" THEN END
370 N=N+1
380 B=N
390 LOCATE 0,CSRLIN-2
400 RETURN

```

### 1.3.2 BERNOUILLI GETALLEN

Bernouilli is de familienaam van een geslacht waarvan we vele telgen in de wis- en natuurkundeboeken van vandaag tegenkomen. Het is daarom redelijk om bij het onderwerp „Bernouilli-getallen” allereerst te vragen „welke Bernouilli?”. Welnu de Bernouilli waarop we doelen was Jacob Bernouilli die leefde van 1654 tot 1705.

Wederom is onze bron het werk van Struik, we citeren: „Jacob begon zijn briefwisseling met Leibniz in 1687. Door een constante uitwisseling van ideeën tussen Leibniz en de twee broeders, de broeders soms in heftige rivaliteit, ontdekten deze drie wiskundigen talloze schatten die door het pionierswerk van Leibniz aan het licht waren gebracht. Het aantal hunner ontdekkingen is groot en bevat vele onderzoeken over integralen en gewone differentiaalvergelijkingen”.

We kunnen hier enige voorbeelden geven. Bij Jacob vinden we het gebruik van poolcoördinaten, de studie van de kettlinglijn (reeds door Huygens en anderen besproken), de lemniscaat (1694) en de logaritmische spiraal.... Jacob Bernouilli was ook onderzoeker van de nog nieuwe waarschijnlijkheidsleer.

Op zijn grafsteen liet hij de logaritmische spiraal graveren met de spreuk: „eadem mutata resurgo”, oftewel: „ofschoon veranderd, ik blijf dezelfde”.

De getallen van Bernouilli hebben de volgende gedaante:

$$B_n = \frac{2(2n)!}{(2^{2n} - 1) \pi^{2n}} \left(1 + \frac{1}{3^{2n}} + \frac{1}{5^{2n}} + \dots\right)$$

$$B_1 = 1/6, B_2 = 1/30, B_3 = 1/42, \dots$$

Deze getallen komen met name in een aantal reeksontwikkelingen voor en ook bij bepaalde numerieke benaderingen van integralen treft men ze aan. Vooral door dit laatste zijn ze van belang bij het werken met computers.

Een uitgebreide verhandeling over deze toepassingen treft men aan in „complements de mathematiques” van Angot (uitgave: Collection technique et scientifique du C.N.E.T.).

*Voorbeeld:*

```
**** Bernouilli ****
```

```
te berekenen Bernouilli getal? 4
```

```
Bernouilli getal 4 : .0333332
```

```
Bernouilli getal 5 : .0757576
```

```
Bernouilli getal 6 : .253114
```

```
Bernouilli getal 7 : 1.16667
```

```
volgende term? (j/n)
```

*Programma:*

```
10 REM *** Bernouilli ***
20 CLS:LOCATE 7,2
30 PRINT"**** Bernouilli ****"
40 REM ** invoer gegevens **
50 GOSUB 140
60 REM ** berekening **
70 GOSUB 200
```

```

80 REM ** uitvoer v/h resultaat **
90 GOSUB 340
100 REM ** volgende term? **
110 GOSUB 370
120 GOTO 60
130 END
140 REM ** invoer gegevens **
150 PI=3.14159265#
160 LOCATE 0,4
170 INPUT"te berekenen Bernouilli getal";N
180 X=2*N
190 RETURN
200 REM ** berekening **
210 REM * faculteitberekening *
220 X2=X^2
230 F=1+1/(12*X)+1/(288*X2)-139/(51840!*X*X2)-571
/(2488320!*X2*X2)
240 R=SQR(2*PI*X)*X^X*EXP(-X)*F
250 REM * berekening factor *
260 F=2*R/((2^X-1)*PI^X)
270 REM * bepaling Bernouilligetel *
280 B1=F:A=1
290 A=A+2
300 B2=B1
310 B1=B2+F/A^X
320 IF (ABS(B2-B1)>(1E-03)) THEN 290
330 RETURN
340 REM ** uitvoer van resultaat **
350 PRINT:PRINT"Bernouilli getal"N:"CSNG(B1)
360 RETURN
370 REM ** volgende term? **
380 PRINT:PRINT"volgende term? (j/n)"
390 A$=INKEY$:IF A$="" THEN 390
400 IF A$="N" OR A$="n" THEN END
410 LOCATE 0,CSRLIN-2
420 N=N+1
430 X=N*2
440 RETURN

```

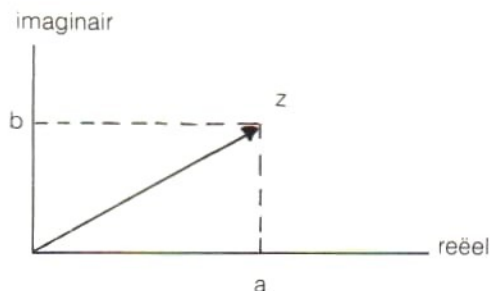
## 1.4 COMPLEXE GETALLEN

Naast de reële getallen kennen we de complexe getallen. Een complex getal ( $z$ ) wordt in het algemeen geschreven als:

$$a + jb$$

Hierin stelt  $a$  het reële en  $b$  het imaginaire gedeelte voor. Verder geldt dat het kwadraat van  $j$  met  $-1$  overeenkomt.

Een complex getal kunnen we steeds als volgt afbeelden:



De lengte van de pijl  $z$  komt overeen met de modulus van  $z$ , deze lengte kan men via de stelling van Pythagoras eenvoudig berekenen. Uit bovengenoemde voorstelling kan men direct afleiden dat complexe getallen ook door middel van poolcoördinaten gegeven kunnen worden. De straal komt dan met de modulus van  $z$  overeen en het argument met  $\arctan(b/a)$ .

In het nu volgende treft men twee programma's aan waarmee de meest voorkomende bewerkingen op complexe getallen kunnen worden uitgevoerd.

### 1.4.1 COMPLEXE GETALLEN 1

Met behulp van dit programma kunnen we van een gegeven complex getal bepalen:

- poolcoördinaten
- e-macht
- natuurlijke logaritme
- wortel

Deze bewerkingen worden op het beeldscherm aangegeven met een letter. Door het intypen van een letter kiest men een bewerking.

Voorbeeld:

```
**** complexe getallen 1 ****

mogelijke bewerkingen:
omzetten naar poolcoördinaten (P)
bepalen van E-macht           (E)
bepalen van LN                 (L)
bepalen van wortel            (W)

bewerking?p

wat is het complexe getal (re,im)
? 3,4

De poolcoördinaten van het
complexe getal ( 3 , 4 ) zijn :

    modulus: 5
    argument: .927295

nog een bewerking? (j/n)
Ok
```

Programma:

```
10 REM *** complexe getallen ***
20 REM ** invoer van complex getal **
30 PI#=3.141592654#
40 KEY OFF:CLS:LOCATE 5,2
50 PRINT"**** complexe getallen 1 ****"
80 PRINT:PRINT
90 PRINT"mogelijke bewerkingen:"
100 PRINT"omzetten naar poolcoördinaten (P)"
110 PRINT"bepalen van E-macht           (E)"
120 PRINT"bepalen van LN                 (L)"
130 PRINT"bepalen van wortel            (W)"
140 PRINT:PRINT"bewerking?";
150 BW$=INKEY$:IF BW$="" THEN 150
160 PRINTBW$
170 IF BW$="P" OR BW$="p" THEN BW=1
180 IF BW$="E" OR BW$="e" THEN BW=2
190 IF BW$="L" OR BW$="l" THEN BW=3
200 IF BW$="W" OR BW$="w" THEN BW=4
```

```

210 IF BW=0 THEN PRINT"verkeerde invoer":GOTO 140
220 GOSUB 300
230 REM ** invoer van bewerking **
240 ON BW GOSUB 350,470,560,640
250 PRINT:PRINT
260 PRINT"nog een bewerking? (j/n)"
270 AN$=INKEY$:IF AN$="" THEN 270
280 IF AN$="j" OR AN$="J" THEN 40
290 END
300 REM ** invoer complex getal **
310 PRINT:PRINT
320 PRINT"wat is het complexe getal (re,im)"
330 INPUT" ";A1,B1
340 RETURN
350 REM ** ►► poolcoördinaten **
360 PRINT:PRINT
370 PRINT"De poolcoördinaten van het"
380 PRINT"complexe getal ("A1","B1") zijn : "
390 R1=SQR(A1*A1+B1*B1)
400 IF A1=0 THEN H1=SGN(B1)*PI/2:GOTO 420
410 H1=ATN(B1/A1)
420 IF A1<0 THEN H1=H1+PI
430 A1=R1
440 B1=H1
450 IF BW=1 THEN PRINT:PRINT"      modulus:"CSNG(A1)
:PRINT"      argument:"CSNG(B1)
460 RETURN
470 REM ** bepalen E-macht **
480 PRINT:PRINT
490 PRINT"de E-macht van het "
500 PRINT"complexe getal ("A1","B1") is:":PRINT
510 R =EXP(A1)
520 A1=R*COS(B1)
530 B1=R*SIN(B1)
540 GOSUB 810' * uitvoer *
550 RETURN
560 REM ** bepaling natuurlijke log **
570 PRINT:PRINT
580 PRINT"de natuurlijke logaritme v/h"
590 PRINT"complexe getal is ("A1","B1") is:":PRIN
T
600 GOSUB 390' * bepaal poolcoörd. *
610 A1=LOG(A1)
620 GOSUB 810' * uitvoer *
630 RETURN

```



```

640 REM ** wortel trekken **
650 REM * invoer macht v/d wortel *
660 PRINT:INPUT"macht van de wortel";N
670 PRINT:PRINT
680 PRINT"de "N"de machts wortels van het"
690 PRINT"complexe getal ("A1", "B1") zijn:":PRINT
700 GOSUB 390' * bepaal poolcoörd. *
710 A2=A1^(1/N)
720 B2=B1
730 REM * uitvoer van het resultaat *
740 FOR K=0 TO N-1
750 B1=(B2+2*PI*K)/N
760 A1=A2*COS(B1)
770 B1=A2*SIN(B1)
780 GOSUB 810' * geef wortel weer *
790 NEXT K
800 RETURN
810 REM ** weergave resultaat **
820 IF B1>0 THEN X=A1:GOSUB 870:PRINT" +I";:X=B1:
GOSUB 870
830 IF B1<0 THEN X=A1:GOSUB 870:PRINT" -I";:X=ABS
(B1):GOSUB 870
840 IF B1=0 THEN X=A1:GOSUB 870
850 PRINT
860 RETURN
870 REM * print format *
880 PRINT USING "###.#####";X;
890 RETURN

```

#### 1.4.2 COMPLEXE GETALLEN: BEWERKINGEN

Met behulp van dit programma kunnen we een aantal bewerkingen met twee complexe getallen ( $z_1$  en  $z_2$ ) uitvoeren:

```

optellen:  $z_1 + z_2$ 
aftrekken:  $z_1 - z_2$ 
vermenigvuldigen:  $z_1 * z_2$ 
delen:  $z_1/z_2$ 
machtsverheffen:  $z_1 \uparrow z_2$ 
worteltrekken:  $z_1 \uparrow (1/z_2)$ 
bepalen logaritme:  $\log z_2$  met basis  $z_1$ 

```

Op het beeldscherm worden deze bewerkingen symbolisch aangegeven.

Voorbeeld:

```
**** complexe getallen 2 ****

mogelijk bewerkingen:

optellen                +
aftrekken               -
vermenigvuldigen       *
delen                   /
machtsverheffen (z1 als grondtal) ^
worteltrekken (z2 als grondtal) W
logarithme van z2 als basis z1 L

re1,im1? 2,3
re2,im2? 1,5.5

bewerking?*

    getal z1: ( 2 , 3 )
    getal z2: ( 1 , 5.5 )

vermenigvuldiging (z1*z2)
geeft als resultaat

    -14.5 +I 14

nog een bewerking? (j/n)
Ok
```

Programma:

```
10 REM *** complexe getallen 2 ***
20 REM ** invoeren van gegevens **
30 GOSUB 180
40 REM ** keuze van bewerking **
50 IF BW$="+" THEN GOSUB 590
60 IF BW$="-" THEN GOSUB 640
70 IF BW$="*" THEN GOSUB 690
80 IF BW$="/" THEN GOSUB 750
90 IF BW$="^" THEN GOSUB 850
100 IF BW$="W" THEN GOSUB 910
110 IF BW$="L" THEN GOSUB 970
120 REM ** uitvoer van resultaat **
130 GOSUB 370
140 PRINT:PRINT:PRINT"nog een bewerking? (j/n)"
150 BW$=INKEY$:IF BW$="" THEN 150
160 IF BW$="j" OR BW$="J" THEN 30
170 END
```

```

180 REM ** invoeren van gegevens **
190 CLS:LOCATE 3,2
200 PRINT"**** complexe getallen 2 ****";:PRINT:P
RINT
210 PRINT:PRINT"mogelijk bewerkingen:":PRINT
220 PRINT"optellen          +"
230 PRINT"aftrekken         -"
240 PRINT"vermenigvuldigen  *"
250 PRINT"delen            /"
260 PRINT"machtsverheffen (z1 als grondtal) ^"
270 PRINT"worteltrekken (z2 als grondtal) W"
280 PRINT"logarithme van z2 als basis z1  L"
290 PRINT:INPUT"re1,im1";A1,B1
300 INPUT"re2,im2";A2,B2
310 PRINT:PRINT"bewerking?";
320 BW$=INKEY$:IF BW$="" THEN 320
330 PRINT BW$ :PRINT:PRINT
340 PRINT"  getal z1: ("A1","B1")"
350 PRINT"  getal z2: ("A2","B2")"
360 PRINT:RETURN
370 REM ** uitvoer van resultaten **
380 PRINT"geeft als resultaat":PRINT
390 PRINT" ";:BUF=A1:GOSUB 1050
400 IF B1>0 THEN PRINT" +I";:BUF=B1:GOSUB 1050
410 IF B1<0 THEN PRINT" -I";:BUF=ABS(B1):GOSUB 10
50
420 RETURN
430 REM ** ►► poolcoördinaten **
440 R1=SQR(A1*A1+B1*B1)
450 IF A1=0 THEN H1=SGN(B1)*PI/2:GOTO 470
460 H1=ATN(B1/A1)
470 IF A1<0 THEN H1=H1+PI
480 A1=R1:B1=H1
490 RETURN
500 REM ** bepaling nat. log. **
510 GOSUB 430' * ►► poolcoörd. *
520 A1=LOG(A1)
530 RETURN
540 REM ** bepaling E-macht **
550 R=EXP(A1)
560 A1=R*COS(B1)
570 B1=R*SIN(B1)
580 RETURN
590 REM ** optellen **
600 PRINT"optellen (z1+z2)"
610 A1=A1+A2

```

```

620 B1=B1+B2
630 RETURN
640 REM ** aftrekken **
650 PRINT"aftrekken (z1-z2)"
660 A1=A1-A2
670 B1=B1-B2
680 RETURN
690 REM ** vermenigvuldigen **
700 IF BW$="*" THEN PRINT"vermenigvuldiging (z1*z
2)"
710 R1=A1*A2-B1*B2
720 B1=A1*B2+A2*B1
730 A1=R1
740 RETURN
750 REM ** delen **
760 IF BW$="/" THEN PRINT"deling (Z1/Z2)"
770 GOSUB 430' * ►► poolcoörd. *
780 SWAP A1,A2
790 SWAP B1,B2
800 GOSUB 430' * ►► poolcoörd. *
810 R1=A2/A1
820 A1=R1*COS(B2-B1)
830 B1=R1*SIN(B2-B1)
840 RETURN
850 REM ** machtsverheffen **
860 PRINT"machtsverheffing (Z1^Z2)"
870 GOSUB 500' * bepaal LN *
880 GOSUB 690' * vermenigvuldigen *
890 GOSUB 540' * bepaal E-macht *
900 RETURN
910 REM ** worteltrekken **
920 PRINT"worteltrekken (Z! ^ (1/Z2))"
930 GOSUB 500' * bepaal LN *
940 GOSUB 750' * delen *
950 GOSUB 540' * bepaal E-macht *
960 RETURN
970 REM ** log met compl. grndtal **
980 PRINT"log. bepaling (grondtal =Z1)"
990 GOSUB 550' * bepaal LN *
1000 SWAP A1,A2
1010 SWAP B1,B2
1020 GOSUB 500' * bepaal LN *
1030 GOSUB 750' * delen *
1040 RETURN
1050 REM ** print format **
1060 PRINT CSNG(BUF);
1070 RETURN

```

## 1.5 E-MACHTEN

Dit programma berekent e-machten voor zeer grote waarden van het argument x.

Voorbeeld:

```
X =? 1000
EXP( 1000 ) = 1.97007 * E 434
```

Programma:

```
10 ' *** E-macht ***
20 ' ** invoer van x **
30 GOSUB 90
40 ' ** berekening **
50 GOSUB 150
60 ' ** uitvoer **
70 GOSUB 200
80 END
90 ' ** invoer van x **
100 CLS:KEY OFF
110 PRINT"**** BEREKENING E-MACHT ****"
120 PRINT
130 INPUT"X =";X
140 RETURN
150 ' ** berekening **
160 C=X/LOG(10)
170 B=INT(C)
180 A=10^(C-B)
190 RETURN
200 ' ** uitvoer **
210 PRINT
220 PRINT"EXP("X") ="CSNG(A)"* E"B
230 RETURN
```

## 1.6 FACULTEITEN

Met behulp van de volgende twee programma's kan men faculteiten berekenen. De faculteit van een integer  $n$  komt overeen met:

$$n \times (n - 1) \times (n - 2) \times \dots \times 2 \times 1$$

Het eerste programma berust op de zogenaamde formule van Stirling, met dit programma kan men als argument ook een niet geheel getal invoeren.

Het tweede programma geldt voor gehele getallen.

### 1.6.1 STIRLING-BENADERING

Zoals hierboven beschreven berekent dit programma de faculteit van een getal, waarbij ook niet-gehele getallen als argument ingevoerd mogen worden.

*Voorbeeld:*

```
**** Stirling-benadering ****
****   van faculteiten   ****

x (x≥1) ? 6.45

6.45 ! = 1698
```

*Programma:*

```
10 REM *** Stirling benadering ***
20 CLS:LOCATE 5,2
30 PRINT"**** Stirling-benadering ****"
40 PRINT"      ****   van faculteiten   ****"
50 REM ** invoer van gegevens **
60 GOSUB 120
70 REM ** berekening **
80 GOSUB 160
90 REM ** uitvoer van resultaat **
100 GOSUB 200
110 END
120 REM ** invoer van gegevens **
130 PI#=3.141592654#
140 PRINT:PRINT:INPUT"x (x≥1) ";X
150 RETURN
```

```

160 REM ** berekening **
170 F=1+1/(12*X)+1/(288*X*X)-139/(51840!*X^3)-571
/(2488320##X^4)
180 R=SQR(2*PI*X)*X^X*EXP(-X)*F
190 RETURN
200 REM ** uitvoer van resultaat **
210 PRINT:PRINT:PRINT X"! =";CSNG(R)
220 PRINT
230 RETURN

```

### 1.6.2 FACULTEIT

Dit programma is algemeen toepasbaar. Het berust op het feit dat we een vermenigvuldiging door een optelling van logaritmen kunnen voorstellen. Het berekent tevens de faculteit van zeer grote getallen.

*Voorbeeld:*

```
**** faculteit ****
```

```
Geef getal >0 ? 100
```

```
100 ! = 9.33262 *E 157
```

*Programma:*

```

10 REM *** FACULTEIT ***
20 REM ** input **
30 CLS:LOCATE 5,2
40 PRINT"**** faculteit ****"
50 PRINT:PRINT:INPUT"Geef getal >0 ";NR
60 REM ** berekening **
70 FOR A=1 TO NR
80 S=S+LOG(A)
90 NEXT A
100 S=S/LOG(10)
110 REM ** uitvoer **
120 SI=S-INT(S)
130 PRINT:PRINT:PRINT NR"! =";CSNG(10^SI);"*E";IN
T(S)
140 END

```

## 1.7 INTEGREREN EN DIFFERENTIËREN

Sinds Newton en Leibnitz behoren integreren en differentiëren stellig tot een van de meest uitgevoerde mathematische bewerkingen.

Het aardige is dat met de computer deze bewerkingen weer tot een vorm worden teruggebracht die direct aanvoelbaar is: integreren komt neer op het verdelen van de functie in reepjes, het oppervlak van die reepjes wordt daarna opgeteld.

Differentiëren komt neer op het bepalen van twee dicht bijeen gelegen functiewaarden, waarna de toename als functie van  $\Delta x$  wordt bekeken.

### 1.7.1 INTEGREREN

Het hier gegeven programma berekent de integraal van een functie tussen twee aangegeven grenzen. De functie zelf wordt gegeven in regel 70. Bij het bepalen van de integraal wordt gebruik gemaakt van de regel van Simpson.

Achtereenvolgens wordt gevraagd naar de integratiegrenzen en de gewenste nauwkeurigheid.

*Voorbeeld:*

Bereken de integraal van  $f(x) = x^2 + \sin(x)$  tussen de grenzen 2 en 3

**\*\*\*\* INTEGREREN \*\*\*\***

De te integreren functie  
is te veranderen door regel 70  
te wijzigen

Default functie :

$f(x) = X^2 + \text{SIN}(X)$

ondergrens? 2  
bovengrens? 3  
Tolerantie? 0.0001

de integraal is : 6.90718

Soms is het mogelijk de functie zelf analytisch te integreren, wat tot nauwkeuriger resultaten kan leiden. In dit geval kunt u ook een indruk krijgen van de nauwkeurigheid van dit programma.



Programma:

```
10 ' **** INTEGREREN ****
20 CLEAR:KEY OFF:CLS:DEFSNG A:AS=500
30 PRINTSPC(8)"**** INTEGREREN ****
40 LOCATE 0,4
50 ' functie invoer
60 F$="X^2+SIN(X)"
70 DEF FNF(X)=EXP(X)
80 PRINT"De te integreren functie
90 PRINT"is te veranderen door regel 70
100 PRINT"te wijzigen
110 PRINT:PRINT"Default functie :
120 PRINT:PRINTSPC(6)"f(x)="F$
130 PRINT
140 INPUT"ondergrens";X0
150 INPUT"bovengrens";X4
160 IF X0<X4 THEN 190
170 PRINT"? ondergrens moet kleiner zijn als
    bovengrens"
180 GOTO 140
190 INPUT"Tolerantie";EPS
200 DIM A(AS,9):J=0:S1=0
210 MA=180*EPS/ABS(X4-X0)
220 F0=FNF(X0):F4=FNF(X4):F2=FNF((X0+X4)/2)
230 GOSUB 290' bereken integraal
240 PRINT
250 IF J<700 THEN PRINT"de integraal is :"
```

```

380 J=J+1:IF J>AS THEN 490
390 X4=X2:F4=F2:F2=F1
400 GOTO 290:'integraal op <x0,x2>
410 S1=S:A(J,9)=2:J=J+1
420 X0=X2:F0=F2:F2=F3
430 GOTO 290'integraal op <x2,x4>
440 S=S+S1:S1=A(J,8)
450 J=J-1:IF J<0 THEN 490
460 X0=A(J,0):X2=A(J,1):X4=A(J,2):F0=A(J,3)
470 F1=A(J,4):F2=A(J,5):F3=A(J,6):F4=A(J,3)
480 ON A(J,9) GOTO 410,440
490 RETURN

```

*Opmerking:*

1. Bij bepaalde, grillig verlopende functies is het mogelijk dat het programma de integraal niet met de door u gevraagde nauwkeurigheid kan bepalen. Probeer het opnieuw met een grotere tolerantie.

### 1.7.2 DIFFERENTIËREN

Het hier beschreven programma bepaalt de afgeleide van een functie in een aan te wijzen punt door rond dat punt twee functiewaarden te bepalen.

De functie is gegeven in regel 70 van de listing.

Stel dat we de afgeleide van de functie  $x^2 + \sin(x)$  willen bepalen.

Het programma vraagt nu om het punt waarvan de afgeleide bepaald moet worden. Als z.g. **delta-x**-waarde is in het programma de waarde 0.002 genomen. Uiteraard hangt zo'n keuze af van de functie waarvan men de afgeleide in een bepaald punt wil benaderen en de nauwkeurigheid die men wenst te bereiken.

Het programma is alleen bedoeld om voor een bepaalde functie snel afgeleiden te bepalen. Een nog betere weg is uiteraard het analytisch differentiëren van de functie zelf. Zo is de afgeleide van de gegeven functie  $2x + \cos(x)$  en voor  $x=5$  vinden we de waarde  $10 + 0.2836622 = 10.2836622$ .

Aan de hand van dergelijke berekeningen kan men tevens een indruk van de nauwkeurigheid van het programma verkrijgen.

Voorbeeld:

Bereken de afgeleide waarde van de functie  $x^2 + \sin(x)$  in het punt  $x = 5$ .

\*\*\*\* DIFFERENTIEREN \*\*\*\*

De te differentiëren functie  
is te veranderen door regel 70  
te wijzigen

Default functie :

$f(x) = X^2 + \text{SIN}(X)$

Voor welke X-waarde wilt u  
differentiëren? 5

Voor  $x = 5$  bedraagt de  
afgeleide : 10.2837

Programma:

```
10 ' **** DIFFERENTIEREN ****
20 CLEAR:KEY OFF:CLS
30 PRINTSPC(6)"**** DIFFERENTIEREN ****
40 LOCATE 0,4
50 ' functie invoer
60 F$="X^2+SIN(X)"
70 DEF FNF(X)=X^2+SIN(X)
80 PRINT"De te differentiëren functie
90 PRINT"is te veranderen door regel 70
100 PRINT"te wijzigen
110 PRINT:PRINT"Default functie :
120 PRINT:PRINTSPC(6)"f(x)="F$
130 ' input x-waarde
140 PRINT
150 PRINT"Voor welke X-waarde wilt u"
160 INPUT"differentiëren";X1
170 ' berekening
180 X=X1-1E-04
190 F1=FNF(X)
200 X=X1+1E-04
210 F2=FNF(X)
220 RES=(F2-F1)/2E-04
```

```

230 ' resultaat
240 PRINT
250 PRINT"Voor x="CSNG(X1)"bedraagt de
260 PRINT"afgeleide  ":"CSNG(RES)
270 PRINT
280 END

```

## 1.8 MATRIXBEWERKINGEN

De meest algemene definitie van een matrix is: een matrix is een rechthoek van getallen. Een dergelijke rechthoek kunnen we dan verdelen in een aantal rijen en een aantal kolommen.

Matrixberekeningen worden in zeer veel situaties toegepast en de lezer die graag een verteerbare inleiding over matrices wil doornemen wordt verwezen naar de Aula-uitgave van W.W. Sawyer: „Wegwijs in de wiskunde”.

In het nu volgende treft men programma's aan voor:

- het vereenvoudigen van matrices
- het bepalen van de determinant
- het oplossen van een stelsel vergelijkingen
- het bepalen van de inverse matrix.

### 1.8.1 VEREENVOUDIGEN VAN MATRICES

Met behulp van dit programma kunnen we een matrix vereenvoudigen.

Het programma vraagt eerst om het aantal kolommen en het aantal rijen. Hierna dienen dan de afzonderlijke getallen van de matrix te worden ingevoerd.

Uiteindelijk wordt de matrix in de volgende vorm gebracht:

```

1 x x x . . . x
0 1 x x . . . x
0 0 1 x . . . x
0 0 0 1 . . . x

```

Hierbij stelt x een willekeurig getal voor.

*Voorbeeld:*

```

**** matrix vereenvoudigen ****

```

```

Aantal kolommen ? 3
Aantal rijen    ? 3

```

geef de elementen van de matrix

```
rij 1 : kolom 1 ? 1
        kolom 2 ? 2
        kolom 3 ? 3
rij 2 : kolom 1 ? 1
        kolom 2 ? 4
        kolom 3 ? 9
rij 3 : kolom 1 ? 1
        kolom 2 ? 8
        kolom 3 ? 27
```

even geduld aub

Vereenvoudigde matrix:

1	2	3
Ø	1	3
Ø	Ø	1

Programma:

```
1Ø REM *** matrix vereenvoudigen ***
2Ø REM ** invoer van matrix **
3Ø GOSUB 24Ø
4Ø REM ** hoofdprogramma **
5Ø K1=1
6Ø FOR R1=1 TO M
7Ø REM * zoek element<>Ø in kol k2 *
8Ø FOR K2=K1 TO N
9Ø FOR R2=R1 TO M
1ØØ IF A(R2,K2)<>Ø THEN K1=K2:GOTO 18Ø
11Ø NEXT R2
12Ø NEXT K2
13Ø REM * vegen beëindigd *
14Ø REM * geef vereenvoudigde
15Ø REM * matrix weer *
16Ø GOSUB 41Ø
17Ø END
18Ø REM * wissel rijen r1 en r2 om *
19Ø GOSUB 62Ø
2ØØ REM * veeg kolom k1 schoon
21Ø GOSUB 68Ø
22Ø NEXT R1
23Ø GOTO 13Ø
24Ø REM * invoer van matrix *
25Ø CLS:LOCATE 3,Ø
```

```

260 PRINT"**** matrix vereenvoudigen ****":LOCATE
  0,4
270 INPUT"Aantal kolommen ";N
280 INPUT"Aantal rijen ";M
290 DIM A(M,N)
300 PRINT
310 PRINT"geef de elementen van de matrix"
320 PRINT
330 FOR I=1 TO M
340 PRINT"rij"I" :";
350 FOR J=1 TO N
360 PRINT TAB(10)"kolom"J;
370 INPUT A(I,J)
380 NEXT J:NEXT I
390 PRINT:PRINT"even geduld aub"
400 RETURN
410 REM * uitvoer vereenv. matrix *
420 PRINT:PRINT"Vereenvoudigde matrix:"
430 PRINT
440 REM * maak elementen op
450 REM * hoofddiagonaal >0 *
460 IF N<M THEN M1=N ELSE M1=M
470 FOR I=1 TO M1
480 IF A(I,I)>=0 THEN 520
490 FOR J=1 TO N
500 A(I,J)=-A(I,J)
510 NEXT J
520 NEXT I
530 REM * druk matrix af *
540 FOR I=1 TO M
550 FOR J=1 TO N
560 PRINT USING"####";A(I,J),
570 NEXT J
580 PRINT
590 NEXT I
600 PRINT
610 RETURN
620 REM * swap rijen r1 en r2 *
630 IF R1=R2 THEN 670
640 FOR I=1 TO N
650 SWAP A(R1,I),A(R2,I)
660 NEXT I
670 RETURN
680 REM * k1 schoonvegen met r1 *
690 FOR R2=R1 TO M
700 IF R2=R1 THEN 760

```

```

710 D=A(R2,K1)
720 IF D=0 THEN 760
730 FOR K2=1 TO N
740 A(R2,K2)=A(R2,K2)*A(R1,K1)-A(R1,K2)*D
750 NEXT K2
760 REM * deel rij r2 door GGD *
770 GOSUB 800
780 NEXT R2
790 RETURN
800 REM * deel rij r2 door GGD *
810 REM zoek kleinste element *
820 KL=0
830 FOR I=1 TO N
840 D=ABS(A(R2,I))
850 IF KL=0 OR KL>D THEN KL=D
860 NEXT I
870 IF KL=0 THEN RETURN
880 ' * bepaal de factoren van KL en
890 ' * controleer of deze ook facto-
900 ' ren zyn v/d andere elementen *
910 GGD=1
920 REM * bepaal factor van KL *
930 P=2
940 MX=INT(SQR(KL))
950 IF P>MX THEN P=KL
960 D=KL/P
970 IF D=INT(D) THEN 1000
980 IF P=2 THEN P=3 ELSE P=P+2
990 GOTO 950
1000 KL=D
1010 ' * controleer of P een gemeen-
1020 ' schap. factor is van rij R2 *
1030 FOR I=1 TO N
1040 D=ABS(A(R2,I))/P
1050 IF D<>INT(D) THEN 1080
1060 NEXT I
1070 GGD=GGD*P
1080 IF KL>1 THEN 940
1090 REM * deel rij R2 door GGD *
1100 FOR I=1 TO N
1110 A(R2,I)=A(R2,I)/GGD
1120 NEXT I
1130 RETURN

```

## 1.8.2 HET BEPALEN VAN DE DETERMINANT

Aan een vierkante matrix wordt een getal toegevoegd: de zogenaamde determinant van die matrix. Bijvoorbeeld voor de matrix:

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}$$

is de determinant:

$$a_1 b_2 - a_2 b_1$$

De determinant is slechts te bepalen van een  $n \times n$  matrix, dus van een matrix die evenveel rijen als kolommen heeft. De determinant is onder andere nodig om te bepalen of een stelsel lineaire vergelijkingen al of niet onafhankelijk is.

*Voorbeeld:*

**\*\*\*\* determinant bepalen \*\*\*\***

Dimensie v/d matrix ? 3

Geef nu de elementen van de matrix

```
rij 1 : kolom 1 ? 1
        kolom 2 ? 2
        kolom 3 ? 3
rij 2 : kolom 1 ? 1
        kolom 2 ? 4
        kolom 3 ? 9
rij 3 : kolom 1 ? 1
        kolom 2 ? 8
        kolom 3 ? 27
```

Even geduld aub

Vereenvoudigde matrix:

```
1   2   3
0   1   3
0   0   1
```

Determinant = 12



Programma:

```
10 REM *** determinant bepalen ***
20 REM ** invoer van matrix **
30 GOSUB 250
40 REM * matrix vereenvoudigen *
50 DET=1
60 K1=1
70 FOR R1=1 TO M
80 REM * zoek element<>0 in kol K2 *
90 FOR K2=K1 TO N
100 FOR R2=R1 TO M
110 IF A(R2,K2)<>0 THEN K1=K2:GOTO 190
120 NEXT R2:NEXT K2
130 REM * vegen beëindigd *
140 ' * geef vereenvoudigde matrix *
150 GOSUB 420
160 ' * berekenen en printen van det *
170 GOSUB 580
180 END
190 ' * swap rijen R1 en R *
200 GOSUB 670
210 REM * veeg kolom K1 schoon
220 GOSUB 740
230 NEXT R1
240 GOTO 130
250 REM * invoer van matrix *
260 CLS:LOCATE 4,0
270 PRINT"**** determinant bepalen ****"
280 LOCATE 0,4
290 INPUT"Dimensie v/d matrix ";N
300 M=N
310 DIM A(M,N)
320 PRINT
330 PRINT"Geef nu de elementen van de matrix":PRI
NT
340 FOR I=1 TO M
350 PRINT"rij"I " :";
360 FOR J=1 TO N
370 PRINT TAB(10)"kolom"J;
380 INPUT A(I,J)
390 NEXT J:NEXT I
400 PRINT:PRINT"Even geduld aub"
410 RETURN
```

```

420 ' * Uitvoer van vereenv. matrix *
430 PRINT:PRINT"Vereenvoudigde matrix:":PRINT
440 ' * maak elementen op hfddiag>0 *
450 FOR I=1 TO M
460 IF A(I,I)>=0 THEN 510
470 FOR J=1 TO N
480 A(I,J)=-A(I,J)
490 NEXT J
500 DET=DET*-1
510 NEXT I
520 REM * druk matrix af *
530 FOR I=1 TO M
540 FOR J=1 TO N
550 PRINT A(I,J);" ";
560 NEXT J:PRINT:NEXT I
570 RETURN
580 ' * berekenen v/d determinant
590 ' mbv de vereenvoudide matrix
600 ' en de correctiefactor DET *
610 FOR I=1 TO N
620 DET=DET*A(I,I)
630 NEXT I
640 ' * uitvoer van determ. waarde *
650 PRINT:PRINT"Determinant =";DET:PRINT
660 RETURN
670 ' * swap R1 en R2 *
680 IF R1=R2 THEN 730
690 FOR I=1 TO N
700 SWAP A(R1,I),A(R2,I)
710 NEXT I
720 DET=DET*-1
730 RETURN
740 ' * kol K1 schoonvegen met R1 *
750 FOR R2=R1 TO M
760 IF R2=R1 THEN 830
770 D=A(R2,K1)
780 IF D=0 THEN 830
790 FOR K2=1 TO N
800 A(R2,K2)=A(R2,K2)*A(R1,K1)-A(R1,K2)*D
810 NEXT K2
820 DET=DET/A(R1,K1)
830 ' * deel rij R2 door GGD *
840 GOSUB 870
850 NEXT R2
860 RETURN

```

```

870 ' * deel rij R2 door GGD *
880 ' * zoek kleinste element *
890 KL=0
900 FOR I=1 TO N
910 D=ABS(A(R2,I))
920 IF (KL=0) OR (KL>D) THEN KL=D
930 NEXT I
940 IF KL=0 THEN RETURN
950 ' * bepaal fact. van KL en kon-
960 ' troleer of deze ook fact. zijn
970 ' van de andere elementen *
980 GGD=1
990 ' * bepaal factor van KL *
1000 P=2
1010 MX=INT(SQR(KL))
1020 IF P>MX THEN P=KL
1030 D=KL/P
1040 IF D=INT(D) THEN 1070
1050 IF P=2 THEN P=3 ELSE P=P+2
1060 GOTO 1020
1070 KL=D
1080 ' * controleer of P een gemeen-
1090 ' sch. factor is van rij R2 *
1100 FOR I=1 TO N
1110 D=ABS(A(R2,I))/P
1120 IF D<>INT(D) THEN 1150
1130 NEXT I
1140 GGD=GGD*P
1150 IF KL>1 THEN 1010
1160 ' * deel rij R2 door GGD *
1170 FOR I=1 TO N
1180 A(R2,I)=A(R2,I)/GGD
1190 NEXT I
1200 DET=DET*GGD
1210 RETURN

```

### 1.8.3 HET OPLOSSEN VAN EEN STELSEL LINEAIRE VERGELIJKINGEN

Dit programma maakt zowel gebruik van het vereenvoudigen van een matrix als van het bepalen van de determinant. Een stelsel lineaire vergelijkingen is uitsluitend onafhankelijk als de determinant van die ( $n \times n$ )-matrix ongelijk 0 is.

De invoer van de matrix moet aan bepaalde eisen voldoen. Elk stelsel vergelijkingen moet namelijk eerst in de volgende vorm geschreven worden:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\
 a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\
 \dots + \dots + \dots + \dots &= \dots \\
 a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m
 \end{aligned}$$

Het zijn de getallen  $a_{ij}$  die moeten worden ingevuld. Het laatste getal  $b_j$  komt steeds op de laatste plaats in de rij invoergegevens.

*Voorbeeld:*

We hebben de volgende vergelijkingen:

$$\begin{aligned}
 x_1 &= 3 - x_2 \\
 x_1 &= 5 - 2x_2
 \end{aligned}$$

Eerst worden ze herschreven:

$$\begin{aligned}
 x_1 + x_2 &= 3 \\
 x_1 + 2x_2 &= 5
 \end{aligned}$$

Hierna wordt het programma gebruikt.

#### \*\*\*\* OPlossen VAN VERGELIJKINGEN

Aantal vergelijkingen ? 2

Geef nu de coëfficiënten van de vergelijkingen:

```

rij 1 : kolom 1 ? 1
        kolom 2 ? 1
        kolom 3 ? 3
rij 2 : kolom 1 ? 1
        kolom 2 ? 2
        kolom 3 ? 5

```

Determinant = 1

Oplossing v/d vergelijkingen:

X 1 = 1

X 2 = 2

Programma:

```
10 ' *** vergelijkingen oplossen ***
20 ' ** invoer van coëfficiënten **
30 GOSUB 260
40 ' ** matrix tereenvoudigen **
50 DET=1
60 K1=1
70 FOR R1=1 TO M
80 ' * zoek element<>0 in kolom K2 *
90 FOR K2=K1 TO N
100 FOR R2=R1 TO M
110 IF A(R2,K2)<>0 THEN K1=K2:GOTO 200
120 NEXT R2:NEXT K2
130 ' * berekenen v/d determinant *
140 GOSUB 1210
150 ' * berekenen van evt snijpunt *
160 GOSUB 1280
170 ' * uitvoer van oplossing *
180 GOSUB 430
190 END
200 ' * swap rijen R1 en R2
210 GOSUB 650
220 ' veeg kolom K1 schoon *
230 GOSUB 720
240 NEXT R1
250 GOTO 130
260 ' * invoer van coëfficiënten *
270 CLS
280 PRINT"**** OPLOSSEN VAN VERGELIJKINGEN
290 LOCATE 0,4
300 INPUT"Aantal vergelijkingen ";M
310 N=M+1
320 DIM A(M,N),B(M,N)
330 PRINT:PRINT"Geef nu de coëfficiënten van de":
PRINT"vergelijkingen:":PRINT
340 FOR I=1 TO M
350 PRINT"rij"I" :";
360 FOR J=1 TO N
370 PRINTTAB(10)"kolom"J;
380 INPUT A(I,J)
390 B(I,J)=A(I,J)
400 NEXT J:NEXT I
410 PRINT:PRINT"Even geduld aub"
420 RETURN
```

```

430 ' * uitvoer van oplossing *
440 CLS
450 PRINT"**** OPLOSSEN VAN VERGELIJKINGEN ****"
460 LOCATE 0,4
470 ' * uitvoer van de coëfficiënten van het stel
sel *
480 PRINT"De coëfficiënten v/h stelsel zijn:":PRI
NT
490 FOR I=1 TO M
500 FOR J=1 TO N
510 PRINT B(I,J) " ";
520 NEXT J:PRINT:NEXT I
530 ' * uitvoer van determinant *
540 PRINT:PRINT"Determinant =";DET
550 ' * uitvoer van oplossing *
560 PRINT
570 IF DET=0 THEN 630
580 PRINT"Oplossing v/d vergelijkingen:":PRINT
590 FOR I=1 TO M
600 PRINT"X" I " ="A(I,N):PRINT
610 NEXT I
620 GOTO 640
630 PRINT"Het stelsel is niet eenduidig oplosbaar
"
640 RETURN
650 ' * swap rijen R1 en R2 *
660 IF R1=R2 THEN 710
670 FOR I=1 TO N
680 SWAP A(R1, I),A(R2, I)
690 NEXT I
700 DET=DET*-1
710 RETURN
720 ' * kolom K1 choonvegen met R1 *
730 FOR R2=1 TO M
740 IF R2=R1 THEN 810
750 D=A(R2, K1)
760 IF D=0 THEN 810
770 FOR K2=1 TO N
780 A(R2, K2)=A(R2, K2)*A(R1, K1)-A(R1, K2)*D
790 NEXT K2
800 DET=DET/A(R1, K1)
810 ' * deel rij R2 door GGD *
820 GOSUB 850
830 NEXT R2
840 RETURN

```

```

850 ' * deel rij R2 door GGD *
860 ' * zoek kleinste element *
870 KL=0
880 FOR I=1 TO N
890 D=ABS(A(R2,I))
900 IF KL=0 OR KL>D THEN KL=D
910 NEXT I
920 IF KL=0 THEN RETURN
930 ' * bepaal de factoren van KL
940 ' en controleer of deze ook
950 ' factoren zijn van de andere
960 ' elementen
970 GGD=1
980 ' * bepaal factor van KL *
990 P=2
1000 MX=INT(SQR(KL))
1010 IF P>MX THEN P=KL
1020 D=KL/P
1030 IF D=INT(D) THEN 1060
1040 IF P=2 THEN P=3 ELSE P=P+2
1050 GOTO 1010
1060 KL=D
1070 ' * controleer of P een gemeen-
1080 ' sch. factor is van rij R2 *
1090 FOR I=1 TO N
1100 D=ABS(A(R2,I))/P
1110 IF D<>INT(D) THEN 1140
1120 NEXT I
1130 GGD=GGD*P
1140 IF KL>1 THEN 1000
1150 ' * deel rij r2 door GGD *
1160 FOR I=1 TO N
1170 A(R2,I)=A(R2,I)/GGD
1180 NEXT I
1190 DET=DET*GGD
1200 RETURN
1210 ' *berekenen v/d determinant
1220 ' mbv de vereenvoudigde matrix
1230 ' en de correctiefactor DET *
1240 FOR I=1 TO M
1250 DET=DET*A(I,I)
1260 NEXT I
1270 RETURN
1280 ' * berekenen v/d de oplossing
1290 ' v/h stelsel *
1300 IF DET=0 THEN 1350

```

```

1310 FOR I=1 TO M
1320 A(I,N)=A(I,N)/A(I,I)
1330 A(I,I)=1
1340 NEXT I
1350 RETURN

```

#### 1.8.4 INVERSE MATRIX

Tot slot een programma om de inverse matrix te bepalen. Deze is alleen te bepalen van een vierkante matrix. Stel, we hebben de volgende matrix:

$$\begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$$

Een matrix vermenigvuldigd met z'n inverse levert per definitie de eenheidsmatrix, dus:

$$\begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} \times \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

waarbij dus  $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$  de inverse matrix is.

Via matrixvereenvoudiging krijgen we twee stelsels vergelijkingen:

$$1a + 2b = 1$$

$$3a + 2b = 0$$

$$1c + 2d = 0$$

$$3c + 2d = 1$$

Als deze vergelijkingen opgelost worden ontstaat de inverse matrix.

*Voorbeeld:*

**\*\*\*\* INVERSE MATRIX BEPALEN \*\*\*\***

Dimensie v/d matrix? 2

Geef de elementen v/d matrix

rij 1 : kolom 1 ? 1  
           kolom 2 ? 2



rij 2 : kolom 1 ? 3  
          kolom 2 ? 2

Even geduld aub

De inverse matrix is:

-.5    .5  
.75   -0.25

*Programma:*

```
10 REM *** inverse matrix bepalen ***
20 REM ** invoer van matrix **
30 GOSUB 240
40 REM ** hoofdprogramma **
50 DET=1
60 FOR R1=1 TO N
70 ' * zoek element<>0 in rij R1 *
80 FOR K1=R1 TO N
90 IF A(R1,K1)<>0 THEN 190
100 NEXT K1
110 IF A(R1,R1)=0 THEN DET=0:GOTO 150
120 NEXT R1
130 ' * correctie vermenigvuldiging *
140 GOSUB 800
150 ' ** einde hoofdprogramma **
160 ' * uitvoer van inverse matrix *
170 GOSUB 470
180 END
190 ' * verwisselen van kolommen *
200 GOSUB 590
210 ' * vegen van rij R1 *
220 GOSUB 680
230 GOTO 110
240 ' * invoer van matrix *
250 CLS:LOCATE 4,0
260 PRINT"**** INVERSE MATRIX BEPALEN ****"
270 LOCATE 0,4
280 INPUT"Dimensie v/d matrix";N
290 DIM A(N,2*N)
300 PRINT:PRINT"Geef de elementen v/d matrix":PRI
NT
310 FOR I=1 TO N
320 PRINT"rij"I" :";
330 FOR J=1 TO N
```

```

340 PRINTTAB(10)"kolom"J;
350 INPUT A(I,J)
360 B(I,J)=A(I,J)
370 NEXT J:NEXT I
380 PRINT:PRINT"Even geduld aub"
390 FOR I=1 TO N
400 FOR J=N+1 TO 2*N
410 A(I,J)=0
420 NEXT J:NEXT I
430 FOR I=1 TO N
440 A(I,N+I)=1
450 NEXT I
460 RETURN
470 ' * uitvoer van inverse matrix *
480 PRINT
490 IF DET=0 THEN 560
500 PRINT"De inverse matrix is:":PRINT
510 FOR I=1 TO N
520 FOR J=N+1 TO 2*N
530 PRINT A(I,J)" ";
540 NEXT J:PRINT:NEXT I
550 GOTO 570
560 PRINT"De inverse matrix bestaat niet"
570 PRINT:PRINT
580 RETURN
590 ' * verwisselen van kol. K1 en
600 ' K2=R1 en van N+K1 en N+K2 *
610 IF K1=R1 THEN 670
620 E=N+K1:F=N+R1
630 FOR I=1 TO N
640 SWAP A(I,K1),A(I,R1)
650 SWAP A(I,E),A(I,F)
660 NEXT I
670 RETURN
680 ' * vegen van rij R1 met kol. k2 *
690 E=N+R1
700 FOR K3=1 TO N
710 IF K3=R1 THEN 780
720 D=A(R1,K3)/A(R1,R1)
730 F=N+K3
740 FOR R2=1 TO N
750 A(R2,K3)=A(R2,K3)-A(R2,R1)*D
760 A(R2,F)=A(R2,F)-A(R2,E)*D
770 NEXT R2
780 NEXT K3
790 RETURN

```

```

800 ' * correctie vermenigvuldiging *
810 IF DET=0 THEN 860
820 FOR I=1 TO N
830 FOR J=1 TO N
840 A(J,N+I)=A(J,N+I)/A(I,I)
850 NEXT J:NEXT I
860 RETURN

```

## 1.9 DE ERRORFUNCTIE

Laten we allereerst de definitie van de errorfunctie geven. Deze luidt als volgt:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du \quad (x > 0)$$

De errorfunctie komt men op allerlei plaatsen in de wiskunde tegen. Allereerst zullen we zien dat er een duidelijke relatie is tussen deze functie en de standaardnormaalverdeling. Ten tweede zullen we zien dat deze functie een belangrijke rol speelt bij allerlei problemen rond warmtegeleiding.

De formule waarmee de **erf(x)** uiteindelijk op de computer benaderd wordt luidt:

$$\operatorname{erf}(x) = 1 - (a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5) e^{-x^2}$$

met  $t = 1/(1 + px)$

en

- $p = 0.3275911$
- $a_1 = 0.254829592$
- $a_2 = -0.284496736$
- $a_3 = 1.421413741$
- $a_4 = -1.453152127$
- $a_5 = 1.061405429$

### 1. Relatie met standaardnormaalverdeling

Om de overeenkomst tussen beide functies te bestuderen beschouwen we allereerst de standaardnormaalverdeling.

Deze wordt weergegeven door:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-1/2 x^2}$$

Voor alle duidelijkheid: het gaat hier om de kansdichtheidsfunctie, met andere woorden: de integraal van deze functie tussen twee aangegeven grenzen geeft de kans op een uitkomst tussen die grenzen, dus:

$$p(x_1 < x < x_2) = \int_{x_1}^{x_2} f(u) \, du$$

Welnu, de relatie tussen kansdichtheidsfunctie en de errorfunctie luidt:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x f^2(u) \, du$$

## 2. Problemen rond warmtegeleiding.

Bij warmtegeleiding kunnen we denken aan een materiaal dat in contact wordt gebracht met een bepaalde temperatuur.

Als de temperatuur van het materiaal nu wordt weergegeven door de functie  $y$  dan zal deze functie afhangen van twee variabelen, namelijk de afstand  $x$  tot de rand waar het in contact staat met een zekere temperatuur en de tijd. Op moment  $t=0$  heeft het blok nog zijn eigen temperatuur en na verloop van tijd zal steeds meer de temperatuur van de omgeving worden aangenomen. De functie  $y$  hangt dus af van twee grootheden, namelijk  $x$  en  $t$ .

$$y = y(x,t)$$

De karakteristieke differentiaalvergelijking die bij het geschetste probleem hoort luidt:

$$\frac{d^2 y}{dx^2} = \frac{1}{k} \frac{dy}{dt} \quad (x > 0, t > 0)$$

Onder de gegeven voorwaarden dat:

$y(0,t) = y_0$  de temperatuur op het randoppervlak blijft de omgevings-temperatuur  $y_0$

en

$y(x,0) = 0$  op moment  $t = 0$  heeft het gehele materiaal de referentietemperatuur 0 (N.B.  $y_0$  is kennelijk het temperatuursverschil!).

luit de oplossing van dit probleem:

$$y(x,t) = y_0 (1 - \operatorname{erf}(x/(2\sqrt{kt})))$$

Voorbeeld:

\*\*\*\* ERRORFUNCTIE \*\*\*\*

X =? 1

ERF( 1 ) = .842701

Programma:

```
10 ' *** ERRORFUNCTIE ***
20 ' ** invoer van x **
30 GOSUB 90
40 ' ** berekenen **
50 GOSUB 150
60 ' ** uitvoer **
70 GOSUB 280
80 END
90 ' ** invoer **
100 CLS:LOCATE 4,0
110 PRINT"**** ERRORFUNCTIE ****"
120 LOCATE 0,4
130 INPUT"X =";X
140 RETURN
150 ' ** berekenen **
160 P=.3275911#
170 A1=.254829592#
180 A2=-.284496736#
190 A3=1.421413741#
200 A4=-1.453152027#
210 A5=1.061405429#
220 IF X<0 THEN X=-X:NEG=1
230 T=1/(1+P*X)
240 H=T*(A1+T*(A2+T*(A3+T*(A4+T*A5))))
250 ERF=1-H*EXP(-X*X)
260 IF NEG=1 THEN X=-X:ERF=-ERF
270 RETURN
```

```

280 ' ** uitvoer **
290 PRINT
300 PRINT"ERF("X") ="CSNG(ERF)
310 PRINT
320 RETURN

```

## 1.10 DE FOURIERTRANSFORMATIE EN HET SCHATTEN VAN SPECTRA

De Fouriertransformatie behoort stellig tot de meest gehanteerde ingenieursgereedschappen die we kennen. Bij een onderzoek naar het meest gerefereerde artikel in de technische wetenschappen, kwamen de klassieke artikelen over deze transformatie als duidelijke winnaar uit de bus.

Deze transformatie stelt ons in staat om problemen uit de „tijd-wereld” te vertalen in problemen uit de „frequentie-wereld”.

Bepaalde veel voorkomende bewerkingen kunnen in de frequentie-wereld veel gemakkelijker uitvallen dan in de tijd-wereld. Men kan hier denken aan het uitvoeren van de zogenaamde convolutie-integraal die in de frequentie-wereld neerkomt op een vermenigvuldiging.

Een van de gebieden waar de Fouriertransformatie bij uitstek van pas komt is het schatten van spectra. Het gaat hierbij om de vraag hoe gemiddeld genomen het vermogen van een signaal binnen een bepaalde frequentieband is.

We raden aan om bij het werken met de computer deze signalen eerst te normeren zodat het gemiddelde 0 en de variantie 1 wordt. Men kan hierbij gebruik maken van de onder 2.2 genoemde programma's. Hierna kent de bepaling van het spectrum doorgaans de volgende stappen.

1. bepaal over een aantal (macht van 2) punten de FFT (Fast Fouriertransformatie).

De zo verkregen grootheid  $X(\omega)$  is een complexe grootheid. Vermenigvuldiging met zijn complex geconjugeerde levert een ruwe schatting van het spectrum.

$$X \cdot X^*$$

Bedenk dat bij de berekening met de computer gebruik wordt gemaakt van twee arrays. Deze arrays stellen respectievelijk het reële en het imaginaire deel voor.

2. Willen we een betere schatting dan wordt de volgende procedure toegepast:

- Transformeer het ruwe spectrum met behulp van de FFT terug; de zo verkregen functie komt met een schatting van de autocorrelatiefunctie overeen.
- Deze functie wordt vermenigvuldigd met een functie, het zogenaamde venster. Een aan te bevelen venster is het venster van Papoulis.
- De zo verkregen gemodificeerde functie wordt teruggedgetransformeerd en levert zo de uiteindelijke schatting van het spectrum.

Voorbeeld:

\*\*\*\* FOURIERTRANSFORMATIE \*\*\*\*

Aantal punten waarover getransform-  
meerd wordt (macht van 2)? 8

Functiewaarden :

1	0
2	.781831
3	.974928
4	.433884
5	-.433884
6	-.974928
7	-.781831
8	8.20584E-10

de Fouriertransformatie

	Re:	Im:
1	0.00000	0.00000
2	1.36930	3.30578
3	-0.62698	-0.62698
4	-0.50153	-0.20774
5	-0.48157	0.00000
6	-0.50153	0.20774
7	-0.62698	0.62698
8	1.36930	-3.30578

Verder gaan? (J/N)

het ruwe spectrum

	Re:	Im:
1	0.00000	0.00000
2	12.80315	0.00000
3	0.78621	0.00000
4	0.29469	0.00000
5	0.23191	0.00000
6	0.29469	0.00000
7	0.78621	0.00000
8	12.80315	0.00000

Verder gaan? (J/N)

de autocorrelatiefunctie

	Re:	Im:
1	28.00000	0.00000
2	17.45771	-0.00000
3	-1.34050	0.00000
4	-17.92154	0.00000
5	-24.39134	0.00000
6	-17.92154	0.00000
7	-1.34050	-0.00000
8	17.45771	-0.00000

Verder gaan? (J/N)

de autocorr fu na Papouliscorr.

	Re:	Im:
1	28.00000	0.00000
2	6.85558	-0.00000
3	-0.00329	0.00000
4	0.00000	0.00000
5	0.00000	0.00000
6	0.00000	0.00000
7	-0.00329	-0.00000
8	6.85558	-0.00000

Verder gaan? (J/N)

uiteindelijke spectrale schatting

	Re:	Im:
1	41.70458	0.00000
2	37.69525	-0.00000
3	28.00657	0.00000
4	18.30475	0.00000
5	14.28227	0.00000
6	18.30475	0.00000
7	28.00657	-0.00000
8	37.69525	-0.00000

Verder gaan? (J/N)



Programma: (gebaseerd op een FORTRAN-programma van Cooley e.a.; Trans. on educ., 1969)

```
10 ' *** FOURIERTRANSFORMATIE ***
20 ' ** begin programma **
30 ' ** te transformeren functie **
40 DEF FN F(T)=SIN(2*PI*(T-1)/(NP-1))
50 WIDTH 37:CLS:LOCATE 2,0
60 PRINT"**** FOURIERTRANSFORMATIE ****"
70 LOCATE 0,3
80 ' * INVOER AANTAL PUNTEN **
90 PRINT:PRINT"Aantal punten waarover getransfor-
meerd wordt (macht van 2)";
100 INPUT NP
110 P=INT(LOG(NP)/LOG(2)+.5)
120 DIM X(NP,2),WI(NP)
130 PI=3.141592654#
140 ' * bepalen fu. in tijddomein *
150 PRINT"Functiewaarden :":PRINT
160 FOR T=1 TO NP
170 X(T,1)=FN F(T)
180 X(T,2)=0
190 PRINT T,CSNG(X(T,1))
200 NEXT T
210 IF P>4 THEN PRINT:PRINT"Even geduld aub":PRIN
T
220 ' ** heentransformatie **
230 TEKEN=1:GOSUB 440' FFT
240 A$=" de Fouriertransformatie"
250 GOSUB 890' uitvoer
260 ' ** ruwe spectrum **
270 GOSUB 1110
280 A$="het ruwe spectrum"
290 GOSUB 890
300 ' ** terugtransformatie **
310 TEKEN=-1:GOSUB 440' FFT
320 A$="de autocorrelatiefunctie"
330 GOSUB 890
340 ' ** Papouliscorrectie **
350 GOSUB 1030' venster maken
360 GOSUB 1170' correctie
370 A$="de autocorr fu na Papouliscorr."
380 GOSUB 890
390 ' ** spectrale schatting
400 TEKEN=1:GOSUB 440' FFT
410 A$="uiteindelijke spectrale schatting"
```

```

420 GOSUB 890
430 END
440 ' ** FFT **
450 N=NP
460 NV2=N/2
470 NM1=N-1
480 J=1
490 FOR I=1 TO NM1
500 IF I>=J THEN 530
510 SWAP X(J,1),X(I,1)
520 SWAP X(J,2),X(I,2)
530 K=NV2
540 IF NOT K<J THEN 590
550 FOR KK=-1 TO 0
560 J=J-K:K=K/2
570 KK=K<J
580 NEXT
590 J=J+K
600 NEXT I
610 FOR L=1 TO P
620 LE=2^L
630 L1=LE/2
640 V1=1
650 V2=0
660 W1=COS(TEKEN*PI/L1)
670 W2=SIN(TEKEN*PI/L1)
680 FOR J=1 TO L1
690 FOR I=J TO N STEP LE
700 IP=I+L1
710 T1=X(IP,1)*V1-X(IP,2)*V2
720 T2=X(IP,1)*V2+X(IP,2)*V1
730 X(IP,1)=X(I,1)-T1
740 X(IP,2)=X(I,2)-T2
750 X(I,1)=X(I,1)+T1
760 X(I,2)=X(I,2)+T2
770 NEXT I
780 Q=V1
790 V1=V1*W1-V2*W2
800 V2=Q*W2+V2*W1
810 NEXT J
820 NEXT L
880 RETURN
890 ' ** uitvoer **
900 PRINT
910 PRINT:PRINT A$:PRINT
920 PRINTTAB(6)"Re:"TAB(22)"Im:":PRINT

```

```

930 FOR K=1 TO NP
940 PRINT K:LOCATE 4,CSRLIN-1:PRINT USING "###.##
###";X(K,1)
950 LOCATE 20,CSRLIN-1
960 PRINT USING "###.####";X(K,2)
970 NEXT K
980 PRINT:PRINTTAB(8)"Verder gaan? (J/N) ";
990 B$=INKEY$:IF B$=""THEN 990
1000 PRINT B$
1010 IF B$="N" OR B$="n" THEN END
1020 RETURN
1030 ' ** papoulisvenster **
1040 N=NP*.4:M=N-1:FA=PI/M
1050 FOR I=2 TO N
1060 T=I-1:AR=FA*T:WI(I)=SIN(AR)/PI+(1-T/M)*COS(A
R)
1070 WI(NP+2-I)=WI(I)
1080 NEXT I
1090 WI(1)=1
1100 RETURN
1110 ' ** ruwe spectrum **
1120 FOR I=1 TO NP
1130 X(I,1)=X(I,1)^2+X(I,2)^2
1140 X(I,2)=0
1150 NEXT I
1160 RETURN
1170 ' ** papouliscorrectie **
1180 FOR I=1 TO NP
1190 X(I,1)=WI(I)*X(I,1)
1200 NEXT I
1210 RETURN

```

### 1.11 VEKTORBEWERKINGEN

De volgende twee programma's vinden hun oorsprong in de vektoranalyse.

Beide programma's maken gebruik van de definitie van het inwendig produkt (kortweg: in-  
produkt) van twee vektoren.

Zijn **a** en **b** twee vektoren, met gelijk aantal elementen (zeg **N**), dan wordt het inproduct van  
deze vektoren, genoteerd als **(a,b)**, gedefinieerd als:

$$(a,b) = \sum_{i=1}^N a_i \cdot b_i$$

### 1.11.1 LENGTE VAN EEN VEKTOR

De lengte van een vektor  $a$ , genoteerd als  $|a|$ , wordt als volgt gedefinieerd:

$$|a| = \sqrt{(a,a)}$$

Het onderstaande programma bepaalt de lengte van een vektor met behulp van bovenstaande definitie.

*Voorbeeld:*

```
**** LENGTE VAN EEN VEKTOR ****
```

```
Aantal elementen? 4
```

```
V[ 1 ] =? 2
```

```
V[ 2 ] =? 3
```

```
V[ 3 ] =? 3
```

```
V[ 4 ] =? 2
```

```
.
```

```
De lengte van de vektor
```

```
bedraagt 5.09902
```

*Programma:*

```
10 ' *** LENGTE VAN EEN VEKTOR ***
20 CLS:LOCATE 0,3
30 PRINT"**** LENGTE VAN EEN VEKTOR ****"
40 LOCATE 0,4
50 ' ** invoer **
60 GOSUB 120
70 ' ** berekening **
80 GOSUB 220
90 ' ** uitvoer **
100 GOSUB 290
110 END
120 ' ** INVOER VAN VEKTOR **
130 PRINT
140 INPUT"Aantal elementen";AE
150 PRINT
160 DIM V(AE)
170 FOR I=1 TO AE
180 PRINT "V["I"] =";
190 INPUT V(I)
```

```

200 NEXT I
210 RETURN
220 ' ** berekening **
230 LL=0
240 FOR I=1 TO AE
250 LL=LL+V(I)*V(I)
260 NEXT I
270 L=SQR(LL)
280 RETURN
290 ' ** uitvoer van lengte **
300 PRINT
310 PRINT"De lengte van de vektor"
320 PRINT
330 PRINT"bedraagt";CSNG(L)
340 PRINT
350 RETURN

```

### 1.11.2 HOEK TUSSEN TWEE VEKTOREN

Voor de hoek  $\phi$  tussen de vektoren  $a$  en  $b$  geldt:

$$\cos \phi = \frac{(a,b)}{|a| |b|}$$

De hoek  $\phi$  ligt tussen de 0 en 180 graden. Houdt er rekening mee dat de hoek tussen de nulvektor (een vektor met alleen de elementen 0) en een willekeurige andere vektor onbepaald is.

*Voorbeeld:*

\*\*\*\* HOEK TUSSEN 2 VEKTOREN \*\*\*\*

Aantal elementen :? 3

Vector 1:

V1[ 1 ] =? 1

V1[ 2 ] =? 2

V1[ 3 ] =? 3

Vector 2:

V2[ 1 ] =? -5

V2[ 2 ] =? 1

V2[ 3 ] =? 1

\*\*\*\* hoek tussen 2 vektoren \*\*\*\*

Vektor 1 :  
1 2 3

Vektor 2 :  
-5 1 1

Lengte van vektor 1 = 3.74166  
Lengte van vektor 2 = 5.19615  
Inprodukt = 0  
Hoek tussen de vektoren = 90 graden

*Programma:*

```
10 ' *** HOEK TUSSEN 2 VEKTOREN ***
20 CLS:LOCATE 4,0
30 PRINT"**** HOEK TUSSEN 2 VEKTOREN ****"
40 LOCATE 0,3
50 ' ** invoer **
60 GOSUB 120
70 ' ** berekening **
80 GOSUB 290
90 ' ** uitvoer **
100 GOSUB 450
110 END
120 ' ** invoer van vektoren **
130 PRINT
140 INPUT"Aantal elementen :";AE
150 PRINT
160 DIM V1(AE),V2(AE)
170 PRINT"Vector 1:"
180 FOR I=1 TO AE
190 PRINT"V1["I"] =";
200 INPUT V1(I)
210 NEXT I
220 PRINT
230 PRINT"Vector 2:"
240 FOR I=1 TO AE
250 PRINT"V2["I"] =";
260 INPUT V2(I)
270 NEXT I
280 RETURN
```

```

290 ' ** berekening **
300 PI=3.141592654#
310 L12=0:L22=0:IPROD=0
320 FOR I=1 TO AE
330 L12=L12+V1(I)*V1(I)
340 L22=L22+V2(I)*V2(I)
350 IPROD=IPROD+V1(I)*V2(I)
360 NEXT I
370 L1=SQR(L12)
380 L2=SQR(L22)
390 IF L1=0 OR L2=0 THEN S=1:GOTO 440
400 IF IPROD=0 THEN H=90:GOTO 440
410 CX=IPROD/(L1*L2)
420 IF ABS(CX)>.99999 THEN H=0:GOTO 440
430 H=90-180*ATN(CX/SQR(1-CX*CX))/PI
440 RETURN
450 ' ** uitvoer **
460 CLS:LOCATE 4,0
470 PRINT"**** hoek tussen 2 vektoren ****"
480 LOCATE 0,4
490 PRINT"Vektor 1 :"
500 FOR I=1 TO AE:PRINT USING "####";V1(I);:NEXT
I
510 PRINT:PRINT
520 PRINT"Vektor 2 :"
530 FOR I=1 TO AE:PRINT USING "####";V2(I);:NEXT
I
540 PRINT:PRINT
550 PRINT"Lengte van vektor 1      =";CSNG(L1)
560 PRINT"Lengte van vektor 2      =";CSNG(L2)
570 PRINT"Inprodukt                =";CSNG(IPROD)
580 PRINT"Hoek tussen de vektoren ";
590 IF S=1 THEN PRINT"is onbepaald"ELSE PRINT"="C
SNG(H)"graden"
600 PRINT
610 RETURN

```

## 1.12 NULPUNTSBEPALING

Dit programma bepaalt het nulpunt van een in de listing in regel 70 gegeven functie, en wel met behulp van de combinatie van twee methoden: de koorden van Newton methode en de bisectiemethode.

Achtereenvolgens voert u in:

- de grenzen (linker- en rechtergrens) waartussen het nulpunt moet liggen,
- de gewenste nauwkeurigheid.

*Voorbeeld:*

Bepaal het nulpunt van de functie  $f(x) = x^2 + \sin(x)$ , liggend tussen -0.1 en -1.

```
**** NULPUNT BEPALEN ***
```

```
De gebruikte functie is te ver-  
anderen door regel 70 te wijzigen.
```

```
Default functie :
```

```
f(x)=X^2+SIN(X)
```

```
linkergrens ? -1  
rechtergrens ? -0.1  
tolerantie ? 0.0001
```

```
Het nulpunt is : -.87667
```

*Programma:*

```
10 ' **** NULPUNT BEPALEN ****  
20 CLEAR:KEY OFF:CLS  
30 PRINTSPC(6)"**** NULPUNT BEPALEN ***  
40 LOCATE 0,4  
50 ' functie invoer  
60 F$="X^2+SIN(X)"  
70 DEF FNF(X)=X^2+SIN(X)  
80 PRINT"De gebruikte functie is te ver-  
90 PRINT"anderen door regel 70 te wijzigen.  
100 PRINT:PRINT"Default functie :  
110 PRINT:PRINTSPC(6)"f(x)="F$
```



```

120 ' invoer van grenzen, tolerantie
130 PRINT
140 INPUT"linkergrens ";A
150 INPUT"rechtergrens ";B
160 INPUT"tolerantie ";EPS
170 EPS=ABS(EPS)
180 ' berekening
190 FA=FNF(A):FB=FNF(B)
200 C=A:FC=FA
210 IF ABS(FC)>=ABS(FB) THEN 230
220 A=B:B=C:C=A:FA=FB:FB=FC:FC=FA
230 M=(B+C)/2
240 IF ABS(M-B)<=EPS THEN 340
250 P=(B-A)*FB
260 IF P>=0 THEN Q=FA-FB ELSE Q=FB-FA:P=-P
270 A=B:FA=FB
280 IF P<=ABS(Q)*EPS THEN 320
290 IF P<(M-B)*Q THEN B=B+P/Q ELSE B=M
300 FB=FNF(B)
310 IF (FB>0)=(FC>0) THEN 200 ELSE 210
320 IF C<B THEN B=B-EPS ELSE B=B+EPS
330 GOTO 300
340 ' uitvoer nulpunt
350 PRINT
360 ZERO=((FB>0)=(FC<0))
370 IF ZERO THEN PRINT"Het nulpunt is : "CSNG(C):
GOTO 390
380 PRINT"Er is geen nulpunt in dit interval.
390 PRINT
400 END

```

*Opmerkingen:*

1. Als in het opgegeven interval meer nulpunten aanwezig zijn, vindt het programma slechts één nulpunt! Eventueel kunt u andere grenzen invoeren om hieromtrent meer informatie te verkrijgen.
2. Als u de zogenaamde tolerantie (gewenste nauwkeurigheid) te klein kiest, kan het programma in een „oneindige lus” terechtkomen ... Kies de tolerantie niet kleiner dan  $1E-7$ .

## 2 STATISTISCHE PROGRAMMA'S

### 2.1 KANSVERDELINGEN

De nu volgende programma's hebben betrekking op kansverdelingen en wel op:

- de binomiaalverdeling
- de Poissonverdeling
- de standaardnormaalverdeling

Deze verdelingen beschrijven bepaalde geïdealiseerde processen.

#### 2.1.1 BINOMIAALVERDELING

Bij de binomiaalverdeling denken we aan experimenten waarbij sprake is van een reeks gebeurtenissen. Laten we, om de gedachte te bepalen, denken aan een dobbelsteen die we tien keer opgooien. Zo'n experiment toont dan tien maal een aantal „ogen". Als we nu bijvoorbeeld de kans willen beziën dat er in zo'n reeks van tien worpen driemaal een „6" optreedt, dan hebben we te maken met een binomiaalverdeling.

Kenmerkend voor die verdeling is namelijk dat de gebeurtenis (in dit geval het gooien van een „6") een constante kans heeft om op te treden ( $p = 1/6$ ).

Volgens de binomiaalverdeling geldt dat de kans, dat in een reeks van 10 ( $n = 10$ ) driemaal ( $x = 3$ ) een „6" met kans  $p = 1/6$  optreedt, overeen komt met:

$$P(x = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

ingevuld:

$$P(x = 3) = \binom{10}{3} \left(\frac{1}{6}\right)^3 \left(\frac{5}{6}\right)^7$$

Op de MSX rekenen we dit als volgt uit:

\*\*\*\* BINOMIALE VERDELING \*\*\*\*

$$P(x=k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Aantal herhalingen  
van het experiment           ? 10  
Kans op succes                ? .166667  
Aantal gewenste successen ? 3

De kans op 3 successen is: .155046

Programma:

```
10 ' *** BINOMIALE VERDELING ***
20 ' ** invoer **
30 GOSUB 90
40 ' ** berekenen van resultaat **
50 GOSUB 220
60 REM ** uitvoer van resultaat **
70 GOSUB 360
80 END
90 ' ** invoer v/d getallen **
100 CLS:LOCATE 4,1
110 PRINT"**** BINOMIALE VERDELING ****"
120 LOCATE 0,4
130 PRINTTAB(15)"n k      n-k"
140 PRINTTAB(7)"P(x=k)=(k)p (1-p)"
150 PRINT:PRINT
160 PRINT"Aantal herhalingen"
170 INPUT"van het experiment      ";N
180 INPUT"Kans op succes          ";P
190 INPUT"Aantal gewenste successen ";K
200 A=K
210 RETURN
220 ' ** berekening **
230 ' * berekening van N over K *
240 H=1
250 K1=K
260 IF K>(N-K) THEN K=N-K
270 FOR I=N TO N-K+1 STEP -1
```

```

280 H=H*I
290 NEXT I
300 FOR I=1 TO K
310 H=H/I
320 NEXT I
330 ' * Berekenen van eigenlijke kans *
340 KANS=H*(P^K1)*((1-P)^(N-K1))
350 RETURN
360 ' ** uitvoer van resultaat **
370 PRINT
380 PRINT"De kans op"A"successen is:"CSNG(KANS)
390 PRINT
400 RETURN

```

### 2.1.2 POISSONVERDELING

De Poissonverdeling staat ook wel bekend als de „verdeling van zeldzame gebeurtenissen”. Om dat in te zien geven we eerst een wat ongewoon voorbeeld. Stel dat we in een stad van ongeveer 100.000 inwoners zomaar een woning binnenvallen met de vraag: „Mevrouw bent u in verwachting?”, dan zal die vraag gewoonlijk met „Nee” beantwoord worden (de deur zal meestal dichtgegooid worden, maar laten we ervan uitgaan dat u schattige babykleertjes bij u hebt...).

Goed dan, de kans ( $p$ ) dat bij een willekeurig huisgezin de ooievaar op bezoek komt is uiteraard zeer gering, maar aan de andere kant is het aantal huisgezinnen ( $n$ ) in die stad zo groot dat het gemiddelde ( $np$ ) toch leidt tot een discreet aantal ( $k = 0, 1, 2, \dots$ ) geboorten per dag. Ziezo....het begrip „verdeling van zeldzame gebeurtenissen” zal nu wel duidelijker zijn geworden.

De Poissonverdeling komen we trouwens in allerlei situaties tegen. We noemen enkele klasieke voorbeelden:

- de kans dat binnen vijf minuten de telefoon gaat.
- de kans dat er in het komende uur in een bepaalde stad 3 ongelukken gebeuren.

De formule die dit alles beschrijft luidt:

$$P(x = k) = \frac{\mu^k e^{-\mu}}{k!}$$

waarbij  $k$  het beoogde aantal „optredens” ( $k = 0, 1, 2, \dots$ ) voorstelt en  $\mu$  het gemiddeld aantal beoogde gebeurtenissen.

Voorbeeld:

In een stad van 100.000 inwoners worden iedere avond gemiddeld 5 baby's ( $\mu = 5$ ) geboren. Wat is nu de kans dat er deze nacht precies 3 baby's geboren worden?

\*\*\*\* POISSON VERDELING \*\*\*\*

$$P(x=k) = \frac{\mu^k e^{-\mu}}{k!}$$

Gewenst aantal successen            ? 3  
Gemiddeld aantal successen (=μ) ? 5  
De kans op 3 successen is .140374

Programma:

```
10 ' *** POISSONVERDELING ***
20 ' ** invoer **
30 GOSUB 90
40 ' ** berekenen van de kans **
50 GOSUB 210
60 ' ** uitvoer **
70 GOSUB 270
80 END
90 ' ** invoer **
100 CLS:LOCATE 5,1
110 PRINT"**** POISSON VERDELING ****"
120 PRINT
130 PRINTTAB(18)"k -μ"
140 PRINTTAB(17)"μ e"
150 PRINTTAB(9)"P(x=k)=—————"
160 PRINTTAB(18)"k!"
170 LOCATE 0,8
180 INPUT"Gewenst aantal successen            ";K
190 INPUT"Gemiddeld aantal successen (=μ) ";MU
200 RETURN
210 ' ** berekenen van de kans **
220 H=EXP(-MU)*MU^K
230 FOR I=1 TO K
240 H=H/I
250 NEXT I
260 RETURN
```

```

270 ' ** uitvoer **
280 PRINT
290 PRINT "De kans op "K" successen is "CSNG(H)
300 PRINT
310 RETURN

```

### 2.1.3 DE STANDAARDNORMAALVERDELING

De standaardnormaalverdeling is stellig een van de meest toegepaste verdelingen in de statistiek. De reden hiervan is „de wet van de grote getallen”, die populair gezegd inhoudt dat als de waarde van een zekere grootheid (lengte van een Nederlander, gewicht van een paardebloem) van zeer veel factoren afhangt, deze waarde normaal verdeeld is met gemiddelde  $\mu$  en standaarddeviatie  $\sigma$ .

Om nu bij dit soort normaalverdelingen uitspraken te kunnen doen, hanteren we de standaardnormaalverdeling, deze wordt wel symbolisch aangegeven met  $N(0,1)$ . Dit houdt in dat het gemiddelde van deze verdeling precies 0 is en de standaarddeviatie komt met 1 overeen. Hierna kunnen we onze te beschouwen grootheden normaliseren,

$$X_{\text{gemodificeerd}} = \frac{X - X_{\text{gemiddeld}}}{\text{standaarddeviatie}}$$

en de „truc” is dan dat deze gemodificeerde waarden standaardnormaal verdeeld zijn. Eerst wat wiskunde. De standaardnormaalverdeling voldoet aan de volgende vergelijking:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-1/2t^2} dt$$

Hierbij geeft  $\phi(x)$  de kans dat een uitkomst  $\leq x$  is.

Voor de berekening op de MSX wordt hierbij gebruik gemaakt van:

$$\phi(x) \sim 1 - \left( \frac{1}{\sqrt{2\pi}} e^{-1/2x^2} \right) (a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5)$$

met  $t = 1/(1 + px)$  en verder

$$\begin{array}{ll}
p = .2316419 & a_3 = 1.781477937 \\
a_1 = .31938153 & a_4 = -1.821255978 \\
a_2 = -.356563782 & a_5 = 1.330274427
\end{array}$$

Voorbeeld: (fictief)

De Zweedse recruten zijn gemiddeld 180 cm lang ( $\mu = 180$ ) met een standaarddeviatie van 10 cm.

Welk percentage zal gemiddeld zijn hoofd stoten bij een deurhoogte van 190 cm?

De gemodificeerde waarde waar het hier om gaat is:

$$\frac{\text{deurhoogte} - \text{gemiddelde hoogte}}{\text{standaarddeviatie}}$$

ingevuld:

$$\frac{190 - 180}{10} = \frac{10}{10} = 1$$

Met behulp van het nu volgende programma vinden we dat de bijbehorende  $\phi$  waarde overeenkomt met 0.841 Dit houdt in dat 84% van de recruten onder de deur door kunnen lopen en tevens dat  $100 - 84 = 16\%$  hun hoofd zullen stoten.

\*\*\*\* STANDAARD NORMAAL VERDELING \*\*\*\*

x =? 1

Kansdichtheid : .24197072450334

Kans op X : .84134474061799

Programma:

```
10 ' *** STANDAARD NORMAAL
20 ' VERDELING ***
30 ' ** invoer **
40 GOSUB 100
50 ' ** berekening **
60 GOSUB 160
70 ' ** uitvoer **
80 GOSUB 320
90 END
100 ' ** invoer **
110 CLS
120 PRINT"**** STANDAARD NORMAAL VERDELING ****"
130 LOCATE 0,4
```

```

140 INPUT "x =";X
150 RETURN
160 ' ** berekening **
170 P=.2316419#
180 A1=.31938153#
190 A2=-.356563782#
200 A3=1.781477937#
210 A4=-1.821255978#
220 A5=1.330274427#
230 PI=3.141592654#
240 H1=SQR(2*PI)
250 H2=(EXP(-X*X/2))/H1
260 IF X<0 THEN X=-X:N=1
270 T=1/(1+P*X)
280 H=T*(A1+T*(A2+T*(A3+T*(A4+T*A5))))
290 FI=1-H2*H
300 IF N=1 THEN X=-X:FI=1-FI
310 RETURN
320 ' ** uitvoer **
330 PRINT
340 PRINT "Kansdichtheid : "H2
350 PRINT
360 PRINT "Kans op X      : "FI
370 PRINT
380 RETURN

```

## 2.2 HISTOGRAM

In de vorige paragrafen werden z.g. ideaalmodellen voor kansverdelingen behandeld. In de praktijk kunnen uiteraard variaties optreden, sterker nog ze treden altijd op. Het hier beschreven programma heeft direct te maken met de praktijk van het verwerken van een serie meetgegevens. Het belangrijkste advies dat we bij het verwerken van een serie meetgegevens kunnen geven is: bepaal eerst een histogram van die gegevens! Het histogram als benadering van de kansdichtheidsverdeling die het statistische proces van de waargenomen gegevens beheerst, is namelijk verreweg de belangrijkste functie die er is. We stellen dit met nadruk, omdat het nogal eens voorkomt dat men bij een serie waarnemingen onmiddellijk tot allerlei berekeningen overgaat. Van zo'n serie gegevens bepaalt men dan gewoonlijk dadelijk het gemiddelde en de standaarddeviatie. Bedenk echter dat grootheden als gemiddelde en standaarddeviatie weinig zeggen als het histogram niet aan bepaalde eisen voldoet. Als we bijvoorbeeld met een histogram met twee toppen te maken hebben (denk maar eens aan de verdeling van leeftijden in een kraamvrouwenkliniek) dan zijn grootheden als gemiddelde en standaarddeviatie domweg misleidend. Voor een verdere verhandeling verwijzen we naar het boek „praktische statistiek met behulp van de pocketcalculator (uitgave SMD/Leiden).



Het nu volgende programma bepaalt van een serie waarnemingen een histogram. Dit vormt dan een directe schatting van de kansdichtheidsfunctie. Voer steeds de volgende procedure uit:

1. bepaal het aantal te gebruiken klassen. Gewoonlijk kiest men als bovengrens voor het aantal klassen  $m$ :

$$m < \sqrt{n}$$

Waarin  $n$  het aantal waarnemingen voorstelt.

2. bepaal op grond van de de waargenomen gegevens zowel een ondergrens als een bovengrens. De ondergrens is gewoonlijk kleiner dan de kleinste waarde van een serie waarnemingen en evenzo is de bovengrens groter dan de grootste waarde. Van deze regel wijkt men alleen af als er duidelijke uitschieters in het waarnemingsmateriaal worden aangetroffen.

#### Voorbeeld:

Bij een onderzoek aan een nieuw soort aardappels vindt men voor de gewichten van een aantal „piepers” de volgende waarden (in grammen):

30, 31, 29, 31, 33, 35, 32, 29, 37, 27

Bepaal van deze gegevens het histogram.

We kiezen als aantal klassen nu de waarde 4.

Als ondergrens kiezen we de waarde 26.5 en als bovengrens de waarde 37.5.

#### Resultaat:

\*\*\*\* HISTOGRAM \*\*\*\*

```
Aantal waarnemingen ? 10
Aantal klassen      ? 4
Ondergrens          ? 26.5
Bovengrens           ? 37.5

Waarneming 1 : ? 30
Waarneming 2 : ? 31
Waarneming 3 : ? 29
Waarneming 4 : ? 31
Waarneming 5 : ? 33
Waarneming 6 : ? 35
Waarneming 7 : ? 32
Waarneming 8 : ? 29
Waarneming 9 : ? 37
Waarneming 10 : ? 27
```

\*\*\*\* HISTOGRAM \*\*\*\*



Aantal waarnemingen = 10  
Ondergrens = 26.5 Bovengrens = 37.5

Programma:

```
10 ' *** HISTOGRAM ***
20 ' ** invoer **
30 GOSUB 80
40 ' ** uitvoer **
50 GOSUB 250
60 IN$=INKEY$:IF IN$="" THEN 60
70 END
80 ' ** invoer **
90 CLS:LOCATE 5,0
100 PRINT"**** HISTOGRAM ****"
110 LOCATE 0,4
120 INPUT"Aantal waarnemingen ";N
130 INPUT"Aantal klassen ";AK
140 INPUT"Ondergrens ";A
150 INPUT"Bovengrens ";B
160 PRINT
170 DIM C(AK+1)
180 FOR K=1 TO N
190 PRINT"Waarneming "K": ";
200 INPUT X
210 J=INT((X-A)/(B-A)*AK)+1
220 IF J>=1 AND J<=AK THEN C(J)=C(J)+1
230 NEXT K
240 RETURN
```

```

250 ' ** uitvoer **
260 CLS:KEY OFF:LOCATE 5,0
270 PRINT"**** HISTOGRAM ****"
280 LU$=CHR$(30)+STRING$(2,29)
290 BLOK$=STRING$(2,219)
300 FOR K=2 TO 18 STEP 2
310 LOCATE 1,20-K
320 PRINT USING "##";K;
330 NEXT K
340 FOR K=1 TO AK
350 LOCATE K*3,20
360 PRINT USING " ##";K;
370 IF C(K)=0 THEN 410
380 FOR J=1 TO C(K)
390 PRINT LU$;BLOK$;
400 NEXT J
410 NEXT K
420 LOCATE 0,22
430 PRINT"Aantal waarnemingen ="N
440 PRINT"Ondergrens ="A;
450 PRINT"Bovengrens ="B;
460 RETURN

```

### 2.3 GEMIDDELDE EN STANDAARDDEVIATIE

Als het histogram er „redelijk symmetrisch” uitziet, is het typeren van de gevonden verdeling aan de hand van een gemiddelde waarde en een standaarddeviatie zinvol.

Bedenk dat de gemiddelde waarde iets zegt over de ligging van de top. Per definitie is deze top gelijk aan de modus. Als we te maken hebben met een scheve verdeling dan zal de gemiddelde waarde uiteraard niet met deze top (modus) samenvallen.

De standaarddeviatie zegt iets over de mate waarin de gevonden waarden variëren en wel ten opzichte van de gemiddelde waarde. Een van de aardigheden van de gekozen formule ter bepaling van de standaarddeviatie is dat deze als schatting precies overeenkomt met de gelijklopende parameter (wortel van de variantie) van de normaalverdeling.

Bedenk dat de formule van de normaalverdeling in feite door maar twee variabelen wordt bepaald, n.l. het gemiddelde en de standaarddeviatie. Dit houdt in, dat als we met een normaal verdeelde populatie te maken hebben, deze twee parameters de volledige verdeling beschrijven.

De gebruikte formules zijn:

$$\text{gemiddelde} = \frac{1}{n} \sum x_i$$

$$\text{standaarddeviatie}^2 = \frac{1}{n-1} (\sum x_i^2 - \frac{1}{n} (\sum x_i)^2)$$

Voorbeeld:

Een computerwinkel verkoopt op 5 achtereenvolgende dagen 4, 6, 5, 7 en 11 MSX-computers. Bepaal het gemiddelde aan computers en tevens de bijbehorende standaarddeviatie.

**\*\* GEMIDDELDE EN STANDAARDDEVIATIE \*\***

Aantal waarnemingen ? 5

Waarneming 1 :? 4  
Waarneming 2 :? 6  
Waarneming 3 :? 5  
Waarneming 4 :? 7  
Waarneming 5 :? 11

Gemiddelde = 6.6  
Standaarddeviatie = 2.70185

Programma:

```
10 ' *** gemiddelde en
20 '      standaarddeviatie ***
30 ' ** invoer **
40 CLS
50 PRINT "** GEMIDDELDE EN STANDAARDDEVIATIE **"
60 LOCATE 0,4
70 INPUT "Aantal waarnemingen ";N
80 PRINT
90 FOR K=1 TO N
100 PRINT "Waarneming "K" :";
110 INPUT X
120 S=S+X
130 SKW=SKW+X*X
140 NEXT K
150 ' ** berekening **
160 M=S/N
170 SD=(SKW-S*S/N)/(N-1)
180 SD=SQR(SD)
190 ' ** uitvoer **
200 PRINT
210 PRINT "Gemiddelde          ="CSNG(M)
220 PRINT "Standaarddeviatie ="CSNG(SD)
230 PRINT
240 END
```

## 2.4 COVARIANTIE EN CORRELATIE-COEFFICIENT

Met dit programma berekent men de covariantie en de correlatie-coëfficiënt. De functie voor de covariantie luidt:

$$s_{xy} = \frac{1}{n-1} (\Sigma xy - \frac{1}{n} (\Sigma x \Sigma y))$$

Letterlijk geeft deze grootheid aan hoe de grootheden  $x$  en  $y$  „tesamen“ variëren. Om een en ander meer tastbaar te maken kunnen we denken aan een onderzoek naar het verband tussen lichaamsgewicht en lengte. We hebben hier te maken met een experiment waarbij iedere proefpersoon leidt tot twee waarnemingen, n.l. een lengte ( $x$ ) en een gewicht ( $y$ ). Kenmerkend voor dit soort studies is dat de vraag gesteld kan worden of er wel een verband bestaat tussen  $x$  en  $y$ . Voor ons voorbeeld: is het zo dat toenemende lengte samenhangt met een toenemend gewicht?

Bovengenoemde formule is het uitgangspunt voor de berekening van de correlatiecoëfficiënt  $r$ . Hiertoe delen we de uitkomst door het produkt van de wortels uit de varianties van  $x$  en  $y$ . De zo gevonden grootheid noemen we de correlatie-coëfficiënt  $r$ :

$$r = \frac{s_{xy}}{s_x s_y}$$

$$\text{met } s_x = \left( \frac{\Sigma x_i^2 - (\Sigma x_i)^2/n}{n-1} \right)^{1/2}$$

$$\text{en } s_y = \left( \frac{\Sigma y_i^2 - (\Sigma y_i)^2/n}{n-1} \right)^{1/2}$$

$n$  = aantal getallenparen

$(x_i, y_i)$  = verschillende getallenparen

Als we de waarnemingen  $x$  en  $y$  in het platte vlak uitzetten dan wordt de vraag „is er verband tussen  $x$  en  $y$ “ gereduceerd tot de vraag „is het op het oog doenlijk om door de gegeven punten een rechte lijn te trekken“.

Welnu, als alle punten precies op een rechte lijn liggen (er is dus een maximaal verband) dan zal de absolute waarde van  $r$  exact 1 zijn. Alle afwijkingen hiervan leiden tot een kleinere waarde. Dit houdt in, dat naarmate  $r$  dicht bij 1 ligt, er sprake is van een groter verband. Zoals altijd schuilen er weer een aantal kleine addertjes onder het gras. Ten eerste moet het aantal punten voldoende groot zijn en ten tweede mag er geen sprake zijn van een aantal „puntenwolken“ in de figuur.

*Voorbeeld:*

Bij een onderzoek naar een nieuwe bloembol vindt men voor gewicht en volume de volgende waarden:

(2,3), (4,7) en (6,10)

Is er verband tussen deze grootheden?

```
**** COVARIANTIE EN COR-  ****
**** RELATIE COEFFICIENT ****
```

```
Aantal getalparen ? 3
```

```
Getalpaar 1  :? 2,3
```

```
Getalpaar 2  :? 4,7
```

```
Getalpaar 3  :? 6,10
```

```
Covariantie          : 7
```

```
Correlatie coefficient : .996616
```

*Opmerking:*

In de praktijk gebruikt men altijd meer waarnemingen om een mogelijk verband na te gaan. We raden aan om met tenminste 30 waarnemingen te werken. Om na te gaan of de gevonden waarde niet van het toeval afhangt kan men de volgende grootheid gebruiken:

$$t = \frac{r\sqrt{(n-2)}}{\sqrt{(1-r^2)}}$$

Indien  $n > 30$  is  $t$  bij benadering standaardnormaal verdeeld en wel onder de vooronderstelling dat tussen  $x$  en  $y$  geen verband bestaat! Indien dat verband er wel is zal  $t$  een significant afwijkende waarde aannemen. Voor een betrouwbaarheid van 5% ligt de grenswaarde voor  $t$  bij 1.65.

*Programma:*

```
10 ' *** COVARIANTIE EN
20 '     CORRELATIE COEFFICIENT ***
30 ' ** invoer **
40 GOSUB 100
50 ' ** berekenen **
60 GOSUB 230
70 ' ** uitvoer **
80 GOSUB 420
```

```

90 END
100 ' ** invoer v/d getalparen **
110 CLS:LOCATE 3,0
120 PRINT"**** COVARIANTIE EN COR- ****"
130 PRINT"    **** RELATIE COEFFICIENT ****"
140 LOCATE 0,4
150 INPUT"Aantal getalparen ";N
160 DIM A(N),B(N)
170 PRINT
180 FOR I=1 TO N
190 PRINT"Getalpaar"I" :";
200 INPUT A(I),B(I)
210 NEXT I
220 RETURN
230 ' ** berekenen **
240 ' ** bepalen covariantie **
250 FOR I=1 TO N
260 AI=A(I)
270 BI=B(I)
280 S1=S1+AI
290 S2=S2+BI
300 S3=S3+AI*BI
310 S4=S4+AI*AI
320 S5=S5+BI*BI
330 NEXT I
340 COV=(S3-S1*S2/N)/(N-1)
350 ' ** bepalen corr. coëfficiënt **
360 SX=(S4-S1*S1/N)/(N-1)
370 SX=SQR(ABS(SX))
380 SY=(S5-S2*S2/N)/(N-1)
390 SY=SQR(ABS(SY))
400 CRR=COV/SX/SY
410 RETURN
420 ' ** uitvoer **
430 PRINT
440 PRINT"Covariantie           : "COV
450 PRINT"Correlatie coëfficiënt : "CSNG(CRR)
460 PRINT
470 RETURN

```

## 2.5 CURVE FITTING

Curve fitting is de Engelse uitdrukking voor het zoeken naar een curve die optimaal bij een gegeven reeks (x,y)-paren past. Het aardige van dit programma is dat het de gebruiker de mogelijkheid biedt om aan te geven welke curve hij door de gegeven punten wil „fitten”. Natuurlijk doet zich de vraag voor welke curve gekozen moet worden. Boerenverstand is zoals altijd hier de beste leidraad; als bijvoorbeeld alle punten op het oog op een rechte lijn liggen dan kiest men uiteraard de rechte lijn. Als u wilt weten hoe goed de gevonden curve is dan kunt u uiteraard de berekende curve naderhand op papier uittekenen. De mogelijke curven zijn:

$$\begin{aligned}Z &= a_0 + a_1x + a_2y & (1) \\y &= ax + b & (2) \\y &= a_0 + a_1x + a_2x^2 & (3) \\y &= cx^a + dx^b & (4) \\y &= ax^b \quad (a > 0) & (5) \\y &= ae^{bx} \quad (a > 0) & (6)\end{aligned}$$

*Voorbeeld:*

Bij een experiment vindt men de volgende waarden: (4,5), (6,7) en (8,9). Pas het programma toe om na te gaan welke rechte lijn bij deze punten past.

**\*\*\*\* CURVE FITTING \*\*\*\***

De volgende curves zijn beschikbaar:

$$\begin{aligned}Z &= A_0 + A_1 * X + A_2 * Y & (1) \\Y &= A * X + B & (2) \\Y &= A_0 + A_1 * X + A_2 * X^2 & (3) \\Y &= C * X^A + D * X^B & (4) \\Y &= A * X^B \quad \text{met } A > 0 & (5) \\Y &= A * \text{EXP}(B * X) \quad \text{met } A > 0 & (6)\end{aligned}$$

Welke curve wilt U benaderen? 2

Aantal punten? 3

U kunt nu de punten invoeren

punt 1 : X,Y =? 4,5  
punt 2 : X,Y =? 6,7  
punt 3 : X,Y =? 8,9

$$\begin{aligned}Y &= A * X^B \quad \text{met } A > 0 & (5) \\Y &= A * \text{EXP}(B * X) \quad \text{met } A > 0 & (6)\end{aligned}$$

Welke curve wilt U benaderen? 2



Aantal punten? 3

U kunt nu de punten invoeren

punt 1 : X,Y =? 4,5

punt 2 : X,Y =? 6,7

punt 3 : X,Y =? 8,9

De punten benaderen curve 2 :

$$Y=A*X+B$$

het best als

$$A= 1$$

$$B= 1$$

Programma:

```
10 ' *** CURVE FITTING ***
20 ' ** invoer v/d curve **
30 GOSUB 110
40 ' ** invoer v/d punten **
50 ON CURVE GOSUB 380,480,480,440,480,480
60 ' ** berekeningen **
70 ON CURVE GOSUB 710,930,970,1030,1090,1140
80 ' ** uitvoer **
90 GOSUB 1180
100 END
110 ' ** invoer **
120 CLS:LOCATE 4,0
130 PRINT"**** CURVE FITTING ****"
140 LOCATE 0,3
150 DIM CUR$(6)
160 CUR$(1)="Z=A0+A1*X+A2*Y (1)"
170 CUR$(2)="Y=A*X+B (2)"
180 CUR$(3)="Y=A0+A1*X+A2*X^2 (3)"
190 CUR$(4)="Y=C*X^A+D*X^B (4)"
200 CUR$(5)="Y=A*X^B met A>0 (5)"
210 CUR$(6)="Y=A*EXP(B*X) met A>0 (6)"
220 ' * overzicht v/d curves *
230 PRINT"De volgende curves zijn beschikbaar:"
240 PRINT
250 FOR I=1 TO 6:PRINTCUR$(I):NEXT I
260 ' ** invoer v/d curve **
270 PRINT
280 INPUT"Welke curve wilt U benaderen";CURVE
```

```

290 IF CURVE<1 OR CURVE>6 THEN LOCATE0,CRSLIN-1:G
OTO 280
300 ' ** invoer van aantal punten **
310 PRINT
320 INPUT"Aantal punten";N
330 DIM X(N),Y(N),Z(N)
340 PRINT:PRINT"U kunt nu de punten invoeren":PRI
NT
350 RETURN
360 RETURN
370 ' ** invoer v/d punten **
380 ' * invoer voor curve 1 *
390 FOR I=1 TO N
400 PRINT"punt"I": X,Y,Z =";
410 INPUT X(I),Y(I),Z(I)
420 NEXT I
430 RETURN
440 ' * invoer voor curve 4 *
450 INPUT"A =";AA1
460 INPUT"B =";BB1
470 PRINT
480 ' * invoer voor curve 2,3,4,5,6 *
490 FOR I=1 TO N
500 PRINT"punt"I": X,Y =";
510 INPUT X(I),Y(I)
520 NEXT I
530 RETURN
540 ' * rekenhulproutines *
550 ' * Z:=Y *
560 FOR I=1 TO N:Z(I)=Y(I):NEXT I
570 RETURN
580 ' * Y:=X^bb1 *
590 FOR I=1 TO N:Y(I)=X(I)^BB1:NEXT I
600 RETURN
610 ' * X:=X^aa1 *
620 FOR I=1 TO N:X(I)=X(I)^AA1:NEXT I
630 RETURN
640 ' * Z:=LOG(Y) *
650 FOR I=1 TO N:Z(I)=LOG(Y(I)):NEXT I
660 RETURN
670 ' * X:=LOG(X) *
680 FOR I=1 TO N:X(I)=LOG(X(I)):NEXT I
690 RETURN
700 ' ** berekeningen **
710 '* curve 1 *
720 FOR I=1 TO N

```

```

730 SX=SX+X(I)
740 SY=SY+Y(I)
750 SZ=SZ+Z(I)
760 S1=S1+X(I)*X(I)
770 S2=S2+Y(I)*Y(I)
780 S3=S3+X(I)*Y(I)
790 S4=S4+X(I)*Z(I)
800 S5=S5+Y(I)*Z(I)
810 NEXT I
820 IF CURVE=4 THEN SX=0:SY=0:SZ=0
830 XX=N*S1-SX*SX
840 XZ=N*S4-SX*SZ
850 IF CURVE=2 OR CURVE>4 THEN A2=0:GOTO 900
860 XY=N*S3-SX*SY
870 YY=N*S2-SY*SY
880 YZ=N*S5-SY*SZ
890 A2=(XX*YZ-XY*XZ)/(XX*YY-XY*XY)
900 A1=(XZ-A2*XY)/XX
910 A0=(SZ-A2*SY-A1*SX)/N
920 RETURN
930 ' * curve 2 *
940 GOSUB 550
950 GOSUB 710
960 RETURN
970 ' * curve 3 *
980 GOSUB 550
990 BB1=2
1000 GOSUB 580
1010 GOSUB 710
1020 RETURN
1030 ' * curve 4 *
1040 GOSUB 550
1050 GOSUB 580
1060 GOSUB 610
1070 GOSUB 710
1080 RETURN
1090 ' * curve 5 *
1100 GOSUB 640
1110 GOSUB 670
1120 GOSUB 710
1130 RETURN
1140 ' * curve 6 *
1150 GOSUB 640
1160 GOSUB 710
1170 RETURN

```

```

1180 ' ** uitvoer van resultaat **
1190 PRINT
1200 PRINT"De punten benaderen curve"CURVE": "
1210 PRINT:PRINTTAB(5);LEFT$(CUR$(CURVE),21)
1220 PRINT:PRINT"het best als":PRINT
1230 ON CURVE GOSUB 1260,1310,1260,1350,1410,1410
1240 PRINT
1250 RETURN
1260 ' * uitvoer curve 1,3 *
1270 PRINTTAB(10)"A0="CSNG(A0)
1280 PRINTTAB(10)"A1="CSNG(A1)
1290 PRINTTAB(10)"A2="CSNG(A2)
1300 RETURN
1310 ' * uitvoer curve 2 *
1320 PRINTTAB(10)"A="CSNG(A1)
1330 PRINTTAB(10)"B="CSNG(A0)
1340 RETURN
1350 :230 * uitvoer curve 4 *
1360 PRINTTAB(10)"A="CSNG(AA1)
1370 PRINTTAB(10)"B="CSNG(BB1)
1380 PRINTTAB(10)"C="CSNG(A1)
1390 PRINTTAB(10)"D="CSNG(A2)
1400 RETURN
1410 ' * uitvoer curve 5,6 *
1420 PRINTTAB(10)"A="CSNG(EXP(A0))
1430 PRINTTAB(10)"B="CSNG(A1)
1440 RETURN

```

## 2.6 COMBINATIES EN PERMUTATIES

Aan het einde van dit hoofdstuk maken we nog een klein uitstapje naar de combinatoriek. Dit is het terrein van de beroepsgokker die nu gewapend met zijn MSX zijn kansen kan wagen.

Met dit programma kunnen we het aantal combinaties en permutaties uitrekenen.

Het aantal combinaties heeft betrekking op de vraag hoeveel rijtjes ter grootte van  $k$  we uit een reeks van  $n$  kunnen samenstellen (hierbij letten we niet op de volgorde). Om de gedachte te bepalen, hoeveel mogelijke rijtjes van 13 kaarten kunnen we uit 52 kaarten samenstellen.

Als we wel op de volgorde letten (m.a.w.  $ab$  is een andere reeks dan  $ba$ ) hebben we te maken met het aantal permutaties.

### *Voorbeeld:*

Op een spijkskaart staan 10 mogelijke gerechten. Hoeveel maaltijden bestaande uit 4 gerechten kan men hieruit samenstellen?

Smakelijk eten!

\*\*\*\* COMBINATIES EN PERMUTATIES \*\*\*\*

Aantal elementen ? 10  
Aantal trekkingen ? 4  
  
Aantal combinaties = 210  
Aantal permutaties = 5040

N.B. Het aantal permutaties vindt men door het aantal combinaties met k! te vermenigvuldigen. Voor het voorbeeld:

$$210 \cdot 4! = 5040$$

Programma:

```
10 ' * COMBINATIES EN PERMUTATIES *
20 GOSUB 60' ** invoer **
30 GOSUB 140' ** berekening **
40 GOSUB 300' ** uitvoer **
50 END
60 ' ** invoer **
70 CLS
80 PRINT"**** COMBINATIES EN PERMUTATIES ****"
90 LOCATE 0,4
100 INPUT"Aantal elementen ";N
110 INPUT"Aantal trekkingen ";K
120 IF K>N THEN PRINT" invoer onjuist":GOTO 100
130 RETURN
140 ' ** berekening **
150 COMB=1
160 IF K>(N-K) THEN K1=N-K ELSE K1=K
170 IF K=0 THEN PERM=1:GOTO 290
180 FOR I=N TO N-K1+1 STEP -1
190 COMB=COMB*I
200 NEXT I
210 PERM=COMB
220 FOR I=1 TO K1
230 COMB=COMB/I
240 NEXT I
250 IF K1=K THEN 290
260 FOR I=K1+1 TO K
270 PERM=@ERM*I
280 NEXT I
290 RETURN
```

```
300 ' ** uitvoer **  
310 PRINT  
320 PRINT"Aantal combinaties ="CSNG(COMB)  
330 PRINT"Aantal permutaties ="CSNG(PERM)  
340 PRINT  
350 RETURN
```

## 3 FINANCIËLE BEREKENINGEN

De nu volgende acht programma's zijn tesamen geschikt om zeer uiteenlopende berekeningen uit te voeren op financieel gebied. Alle programma's hebben dezelfde formule als uitgangspunt, namelijk:

$$EW = CW \cdot (1 + i)^n$$

met  $EW$  = eindwaarde  
 $CW$  = beginwaarde (constante waarde)  
 $i$  = peruaage (rentepercentage/100)  
 $n$  = looptijd

### 3.1 BEPALEN VAN TOTAAL GESPAARD BEDRAG

Sparen kan op vele manieren. In dit programma wordt een veel voorkomende manier van sparen behandeld: periodiek een vast bedrag sparen.

De basisformule luidt:

$$\begin{aligned} EW_1 &= PB \times (1 + i)^n \\ EW_2 &= PB \times (1 + i)^{n-1} \\ EW_3 &= PB \times (1 + i)^{n-2} \\ &\dots \\ &\dots \end{aligned}$$

---

$$EW_{\text{totaal}} = PB \times \sum_{T=1}^n (1 + i)^T$$

met  $PB$  = periodiek bedrag

Dit programma berekent wat het eindsaldo is bij gegeven peruaage ( $i$ ) en periodiek bedrag ( $PB$ ) bij gegeven looptijd ( $T$ ).

Opmerking: we mogen ook een looptijd opgeven die met een niet geheel aantal jaren overeenkomt. Om de rente te kunnen omrekenen gebruiken we dan de volgende formule:

$$1 + i = \sqrt[p]{(1 + R/100)}$$

met  $p$  = aantal perioden in een jaar  
 $R$  = rentepercentage op jaarbasis  
 $i$  = perage per periode

Let wel: als de rente op jaarbasis 12% is, is deze op maandbasis géén 1%, maar iets minder (n.l. 0,95%)

*Voorbeeld:*

Een student ontvangt f600,- per maand en wil van dit bedrag f100,- per maand sparen. De geplande studieduur bedraagt 4 jaar (48 maanden). Hoeveel heeft hij na die periode gespaard bij een rentepercentage van 6%?

```

**** BEPALING VAN TOTAAL ****
**** GESPAARD BEDRAG      ****

Periodiek bedrag  Fl ? 100
Aantal perioden   : ? 48
Rente per jaar (%) : ? 6
Perioden per jaar : ? 12

Eindbedrag        : Fl  5418.63
Ontvangen rente  : Fl   618.63
  
```

*Programma:*

```

10 ' *** BEPALING VAN TOTAAL
    GESPAARD BEDRAG ***
20 ' ** invoer gegevens **
30 GOSUB 90
40 ' ** berekening eindbedrag en
    ontvangen rente **
50 GOSUB 190
60 ' ** uitvoer **
70 GOSUB 280
80 END
90 ' ** invoer gegevens **
100 CLS:LOCATE 4,1
110 PRINT"**** BEPALING VAN TOTAAL ****"
120 PRINT"      **** GESPAARD BEDRAG      ****"
130 LOCATE 0,4
140 INPUT"Periodiek bedrag  Fl ";PB
  
```



```

150 INPUT"Aantal perioden      : ";N
160 INPUT"Rente per jaar (%)  : ";R
170 INPUT"Perioden per jaar   : ";P
180 RETURN
190 ' ** berekening eindbedrag en
      ontvangen rente **
200 I=(1+R/100)^(1/P)
210 F=0
220 FOR T=1 TO N
230 F=F+I^T
240 NEXT T
250 EB=PB*F' * eindbedrag *
260 OI=EB-N*PB' *ontvangen rente *
270 RETURN
280 ' ** uitvoer **
290 PRINT
300 PRINT USING "Eindbedrag      : F1 #####.##";E
B
310 PRINT USING "Ontvangen rente : F1 #####.##";O
I
320 PRINT
330 RETURN

```

### 3.2 BEPALEN VAN PERIODIEK BEDRAG

Dit programma is vrijwel identiek aan het vorige programma, het verschil ligt in het feit dat nu een andere grootheid wordt uitgerekend, namelijk het periodiek bedrag bij een gegeven eindwaarde.

#### Voorbeeld:

Dezelfde student wil in plaats van f5418.63 nu precies f6000,- sparen. Wederom gaan we uit van een studieduur van vier jaar (48 maanden), de vraag is nu hoe groot de maandelijkse inleg moet zijn.

```

**** BEPALING VAN PERIODIEK ****
****   TE STORTEN BEDRAG   ****

Gewenst eindbedrag F1 ? 6000
Aantal perioden      : ? 48
Rente per jaar (%)   : ? 6
Perioden per jaar    : ? 12

Periodiek bedrag : F1      110.73
Ontvangen rente  : F1      685.00

```

Programma:

```
10 ' *** BEPALING VAN PERIODIEK
    TE STORTEN BEDRAG ***
20 ' ** invoer gegevens **
30 GOSUB 90
40 ' ** berekenen periodiek bedrag **
50 GOSUB 190
60 ' ** uitvoer **
70 GOSUB 280
80 END
90 ' ** invoer gegevens **
100 CLS:LOCATE 3,1
110 PRINT"**** BEPALING VAN PERIODIEK ****"
120 PRINT"      ****   TE STORTEN BEDRAG      ****"
130 LOCATE 0,4
140 INPUT"Gewenst eindbedrag F1 ";EB
150 INPUT"Aantal perioden      : ";N
160 INPUT"Rente per jaar (%)   : ";R
170 INPUT"Perioden per jaar    : ";P
180 RETURN
190 ' ** berekenen periodiek bedrag **
200  $I=(1+R/100)^{(1/P)}$ 
210 F=0
220 FOR T=1 TO N
230 F=F+I^T
240 NEXT T
250 PB=EB/F' * periodiek bedrag *
260 OI=EB-N*PB' * ontvangen rente *
270 RETURN
280 ' ** uitvoer **
290 PRINT
300 PRINT USING "Periodiek bedrag : F1 ####.#"
;PB
310 PRINT USING "Ontvangen rente   : F1 ####.#"
;OI
320 PRINT
330 RETURN
```

### 3.3 KLIMSPAREN

Banken bedenken steeds weer nieuwe spaarvormen om publiek te trekken. Een spaarvorm die de laatste tijd veel opgang maakt, is het zogenaamde klimsparen. Hierbij wordt een vast bedrag (meestal f 1000, =) gedurende een aantal jaren uitgezet tegen een rente die elk jaar toeneemt. De (dagelijkse) opvraagbaarheid van het uitgezette bedrag varieert van geval tot geval. Het nu volgende programma stelt ons in staat de aanbiedingen van de diverse banken met elkaar te vergelijken.

*Voorbeeld:*

Een bank heeft de volgende aanbieding; de looptijd is 6 jaar en in het eerste jaar is het rentepercentage 4.5%. Ieder jaar neemt dit met 0.5% toe. Wat is nu het werkelijke rentepercentage?

\*\*\*\* KLIMSPAREN \*\*\*\*

```
Beginbedrag (F1) ? 1000
Looptijd (jaar) ? 6

Rente in jaar 1 (%) ? 4.5
Rente in jaar 2 (%) ? 5
Rente in jaar 3 (%) ? 5.5
Rente in jaar 4 (%) ? 6
Rente in jaar 5 (%) ? 6.5
Rente in jaar 6 (%) ? 7

Eindbedrag      : F1   1398.29
Gemiddelde rente: F1     5.75
```

*Programma:*

```
10 ' *** KLIMSPAREN ***
20 ' ** invoer gegevens **
30 GOSUB 90
40 ' ** berekening **
50 GOSUB 220
60 ' ** uitvoer **
70 GOSUB 300
80 END
90 ' ** invoer gegevens **
100 CLS:LOCATE 3,1
110 PRINT"**** KLIMSPAREN ****"
120 LOCATE 0,4
130 INPUT"Beginbedrag (F1) ";B
140 INPUT"Looptijd (jaar) ";L
```

```

150 PRINT
160 DIM R(L)
170 FOR I=1 TO L
180 PRINT "Rente in jaar ";I;
190 INPUT " (% ) ";R(I)
200 NEXT I
210 RETURN
220 ' ** berekening **
230 T=1
240 FOR I=1 TO L
250 T=T*(1+R(I)/100)
260 NEXT I
270 EB=B*T
280 G=(T^(1/L)-1)*100
290 RETURN
300 ' ** uitvoer **
310 PRINT
320 PRINT USING "Eindbedrag          : F1 #####.##";
EB
330 PRINT USING "Gemiddelde rente: F1 #####.##";
G
340 PRINT
350 RETURN

```

### 3.4 PERIODIEKE UITKERING

In plaats van periodiek sparen kan men ook periodiek geld opnemen. We kunnen ons bijvoorbeeld afvragen hoe groot het beginkapitaal moet zijn om gedurende een bepaald aantal perioden, bij een bepaald rentepercentage een vaste uitkering te krijgen. De uitgangformule is nu:

$$BK = PU \times \sum_{T=1}^n \frac{1}{(1+i)^T}$$

met BK = beginkapitaal  
 PU = periodieke uitkering

#### *Voorbeeld:*

De ouders van de student uit de vorige voorbeelden vragen zich af hoeveel geld ze op een rekening zouden moeten zetten om hun zoon elke maand f600,- te kunnen uitkeren.

\*\*\*\* PERIODIEKE UITKERING \*\*\*\*

Periodieke uitkering Fl ? 600  
Aantal perioden : ? 48  
Rente per jaar (%) : ? 6  
Perioden per jaar : ? 12

Beginkapitaal : Fl 25627.63  
Ontvangen rente : Fl 3172.37

Programma:

```
10 ' *** PERIODIEKE UITKERING ***
20 ' ** invoer gegevens **
30 GOSUB 90
40 ' ** berekening **
50 GOSUB 180
60 ' ** uitvoer **
70 GOSUB 270
80 END
90 ' ** invoer gegevens **
100 CLS:LOCATE 3,1
110 PRINT"**** PERIODIEKE UITKERING ****"
120 LOCATE 0,4
130 INPUT"Periodieke uitkering Fl ";PU
140 INPUT"Aantal perioden : ";N
150 INPUT"Rente per jaar (%)" : ";R
160 INPUT"Perioden per jaar : ";P
170 RETURN
180 ' ** berekening beginkapitaal
    en ontvangen rente **
190 I=(1+R/100)^(1/P)
200 F=0
210 FOR T=1 TO N
220 F=F+1/(I^T)
230 NEXT T
240 BK=PU*F' * periodiek bedrag *
250 OI=N*PU-BK' * ontvangen rente *
260 RETURN
270 ' ** uitvoer **
280 PRINT
290 PRINT USING "Beginkapitaal : Fl #####.##";
BK
300 PRINT USING "Ontvangen rente : Fl #####.##";
OI
310 PRINT
320 RETURN
```

### 3.5 PERSOONLIJKE LENING

Dit programma is in principe het tegenovergestelde van het vorige programma. We gaan hier van de volgende situatie uit: iemand leent een bepaald bedrag en betaalt periodiek met vaste bedragen terug. Als deze bedragen gegeven zijn, kunnen we de werkelijke rente uitrekenen.

*Voorbeeld:*

Een zeer alledaags voorbeeld is het kopen op afbetaling. Stel: iemand koopt een televisie van f2000,- en betaalt een jaar lang maandelijks f200,-; wat is nu het werkelijke rentepercentage?

\*\*\*\* PERSOONLIJKE LENING \*\*\*\*

Geleend bedrag	F1 ? 2000
Aflossing per periode	F1 ? 200
Aantal perioden	: ? 12
Perioden per jaar	: ? 12
Rente per jaar :	41.31 %
Betaalde rente : F1	400.00

*Programma:*

```
10 ' *** PERSOONLIJKE LENING ***
20 ' ** invoer gegevens **
30 GOSUB 90
40 ' ** berekening **
50 GOSUB 180
60 ' ** uitvoer **
70 GOSUB 330
80 END
90 ' ** invoer gegevens **
100 CLS:LOCATE 3,1
110 PRINT"**** PERSOONLIJKE LENING ****"
120 LOCATE 0,4
130 INPUT"Geleend bedrag          F1 ";CW
140 INPUT"Aflossing per periode F1 ";PU
150 INPUT"Aantal perioden          : ";N
160 INPUT"Perioden per jaar       : ";P
170 RETURN
```

```

180 ' ** berekening **
190 F=CW/PU
200 OG=0
210 BG=1
220 BF=0
230 I=(OG+BG)/2
240 IF (BG-OG)<1E-05 THEN 300
250 FOR T=1 TO N
260 BF=BF+1/(1+I)^T
270 IF BF>F THEN OG=I:GOTO 220
280 NEXT T
290 BG=I:GOTO 220
300 R=((1+I)^P-1)*100' * rente/jaar *
310 OI=(N*PU)-CW' * betaalde rente *
320 RETURN
330 ' ** uitvoer **
340 PRINT
350 PRINTUSING"Rente per jaar :      ##.## %";R
360 PRINTUSING"Betaalde rente : F1 #####.##";OI
370 PRINT
380 RETURN

```

### 3.6 ANNUITEITENHYPOTHEEK

Bij deze vorm van hypotheek wordt periodiek een vast bedrag afbetaald: de zogenaamde annuïteit. De annuïteit bestaat steeds uit een rentedeel en een aflossingsdeel. Het rentedeel is het nog niet afgeloste bedrag vermenigvuldigd met het rentepercentage. De gebruikte formule is dezelfde als in 3.4 en 3.5, maar nu wordt het af te lossen bedrag **per periode** bepaald.

*Voorbeeld:*

Iemand leent f10.000,- op annuïteitenbasis met een looptijd van 5 jaar. Als de rente 8% is, wat is dan de annuïteit en hoe is dit bedrag over de 5 jaren samengesteld?

**\*\*\*\* ANNUITEITEN-HYPOTHEEK \*\*\*\***

```

Hypotheek           F1 ? 10000
Looptijd (in jaren) : ? 5
Rente per jaar (%)  : ? 8

Annuïteit          : F1 2504.56
Betaalde rente     : F1 2522.62

```

jaar	rente	aflossing
1	800.00	1704.56
2	663.63	1840.93
3	516.36	1988.20
4	357.30	2147.26
5	185.52	2319.04

Programma:

```

10 ' *** ANNUITEITEN-HYPOTHEEK ***
20 ' ** invoer gegevens **
30 GOSUB 90
40 ' ** berekening **
50 GOSUB 170
60 ' ** uitvoer **
70 GOSUB 260
80 END
90 ' ** invoer gegevens **
100 CLS:LOCATE 3,1
110 PRINT"**** ANNUITEITEN-HYPOTHEEK ****"
120 LOCATE 0,4
130 INPUT"Hypothek          F1 ";CW
140 INPUT"Looptijd (in jaren) : ";N
150 INPUT"Rente per jaar (%) : ";R
160 RETURN
170 ' ** berekening **
180 I=R/100
190 F=0
200 FOR T=1 TO N
210 F=F+1/(1+I)^T
220 NEXT T
230 AN=CW/F' * annuïteit *
240 OI=N*AN-CW' * betaalde rente *
250 RETURN
260 ' ** uitvoer **
270 PRINT
280 PRINTUSING"Annuïteit      : F1 #####.##";AN
290 PRINTUSING"Betaalde rente : F1 #####.##";OI
300 PRINT:PRINT
310 PRINT"jaar          rente          aflossing"
320 PRINT
330 FOR T=1 TO N
340 RD=I*CW
350 AD=AN-RD

```



```

360 CW=CW-AD
370 PRINTUSING" ##";T;
380 PRINTUSING" #####.##";RD;
390 PRINTUSING" #####.##";AD
400 NEXT T
410 PRINT
420 RETURN

```

*Opmerking:*

Bij het afsluiten van een hypotheek gelden doorgaans een aantal renteverhogende bepalingen. In het algemeen zijn er vier mogelijkheden van betalen. Van „gunstigste” naar „ongunstigste” wijze van betalen zijn dat: halfjaarlijks achteraf, per kwartaal achteraf, maandelijks achteraf of maandelijks vooraf. Verder dient men met de afsluitprovisie rekening te houden.

Gaan we bijvoorbeeld uit van een provisie van 1,5% en een betaling maandelijks vooraf, dan kunnen we op de nu volgende manier de werkelijke rente berekenen met behulp van het „persoonlijke lening-programma”.

Eerst bepalen we het werkelijk geleende bedrag:

provisie:  $10.000 \cdot 0,015 = 150$  gulden  
 aflossing eerste maand:  $(2504,56)/12 = 208,67$  gulden  
 Deze twee bedragen opgeteld is 358,67 gulden.

In werkelijkheid lenen we dus slechts  $10.000 - 358,67 = 9641,33$ . We moeten nog  $60 - 1 = 59$  maanden aflossen. Geven we deze getallen op aan het „persoonlijke lening-programma”, dan blijkt de werkelijke rente ruim 2,5% hoger.

*N.B. Deze berekening vergt tamelijk veel tijd.*

### 3.7 LINEAIRE HYPOTHEEK

Bij een lineaire hypotheek is het jaarlijks te betalen aflossingsdeel over de gehele looptijd constant. Het rentedeel neemt echter ieder jaar af, waardoor het jaarlijks te betalen bedrag aan rente en aflossing ook afneemt.

In formule:  $\text{rentedeel } n = (\text{HB} - (A \cdot (n - 1)) \cdot i)$

met  $n$  = index voor n-de jaar  
 HB = hypotheekbedrag  
 A = aflossingsdeel  
 i = perUAGE

Bij een variant van de lineaire hypotheek neemt het aflossingsdeel toe, naarmate de hypotheek vordert (hypotheek met klimmende aflossing).

Bijvoorbeeld:

1e periode van 5 jaar	2%
2e periode van 5 jaar	2,5%
3e periode van 5 jaar	3%
4e periode van 5 jaar	3,5%
5e periode van 5 jaar	4%
6e periode van 5 jaar	5%

Beide mogelijkheden zijn met dit programma te berekenen.

**Voorbeeld:**

Iemand leent f 150.000,- op basis van een lineaire hypotheek. Over de eerste 5 jaar betaalt het aflossingsdeel 2% dus f 3000,- per jaar. Het rentepercentage is 8%. Het aflossingsschema voor de eerste periode is dan:

**\*\*\*\* LINEAIRE HYPOTHEEK \*\*\*\***

Hypotheek	F1 ? 150000
Looptijd (in jaren)	: ? 5
Rente per jaar (%)	: ? 8
Aflossingsdeel	F1 ? 3000

jaar	rente	aflossing
1	12000.00	15000.00
2	11760.00	14760.00
3	11520.00	14520.00
4	11280.00	14280.00
5	11040.00	14040.00

**Resthypotheek : F1 135000.00**

Net als voor de annuïteitenhypotheek gelden ook voor de lineaire hypotheek dezelfde renteverhogende bepalingen. In de volgende paragraaf wordt uitgelegd hoe men het werkelijk betaalde rentepercentage kan berekenen.

Programma:

```
10 ' *** LINEAIRE HYPOTHEEK.***
20 ' ** invoer gegevens **
30 GOSUB 70
40 ' ** berekeningen en uitvoer **
50 GOSUB 160
60 END
70 ' ** invoer **
80 CLS:LOCATE 3,0
90 PRINT"**** LINEAIRE HYPOTHEEK ****"
100 LOCATE 0,4
110 INPUT"Hypothek           F1 ";CW
120 INPUT"Looptijd (in jaren) : ";N
130 INPUT"Rente per jaar (%)  : ";R
140 INPUT"Aflossingsdeel      F1 ";AD
150 RETURN
160 ' ** berekeningen en uitvoer *
170 I=R/100
180 PRINT
190 PRINT
200 PRINT"jaar           rente           aflossing"
210 PRINT
220 FOR T=1 TO N
230 RD=I*CW
240 AN=RD+AD
250 CW=CW-AD
260 PRINT USING "  ##";T;
270 PRINT USING "           #####.##";RD;
280 PRINT USING "           #####.##";AN
290 NEXT T
300 PRINT
310 PRINT USING"Resthypothek : F1 #####.##";CW
320 PRINT
330 RETURN
```

### 3.8 RENTABILITEIT

Een belangrijk criterium om een investering te beoordelen is de rentabiliteit van die investering. De rentabiliteit is onmogelijk met de hand te berekenen, dit programma doet het in enkele seconden.

#### Voorbeeld 1:

Een ondernemer moet beslissen of hij zal investeren in project A of project B. Beide projecten vergen een investering van f50.000,- en voor beide projecten is de looptijd 5 jaar. Project A levert gedurende die 5 jaar de volgende winsten op:

1e jaar: 5.000,-  
2e jaar: 10.000,-  
3e jaar: 15.000,-  
4e jaar: 20.000,-  
5e jaar: 20.000,-  
  
totaal: 70.000,-

Project B levert ook totaal 70.000 gulden op, maar de winsten zijn anders over de jaren verdeeld:

1e jaar: 10.000,-  
2e jaar: 20.000,-  
3e jaar: 20.000,-  
4e jaar: 10.000,-  
5e jaar: 10.000,-  
  
totaal: 70.000,-

Met behulp van het programma berekenen we de rentabiliteit van project A:

\*\*\*\* RENTABILITEIT \*\*\*\*

\*\*\*\* RENTABILITEIT \*\*\*\*

Investering	: Fl	50000.00	Investering	: Fl	50000.00
periode		winst (Fl.)	periode		winst (Fl.)
1		5000.00	1		10000.00
2		10000.00	2		20000.00
3		15000.00	3		20000.00
4		20000.00	4		10000.00
5		20000.00	5		10000.00
Rentabiliteit	:	10.10 %	Rentabiliteit	:	12.94 %

Deze bedraagt 10,1%

Met behulp van hetzelfde programma blijkt de rentabiliteit van project B echter 12,9%. Dit is beduidend beter dan project A, dus de ondernemer zal in project B investeren.

#### Voorbeeld 2:

Hoewel het programma primair bedoeld is om de rentabiliteit te berekenen, is dit programma ook zeer geschikt om het werkelijke rentepercentage te bepalen bij een lineaire hypotheek. We nemen hetzelfde voorbeeld als in paragraaf 3.7.

De afsluitprovisie bedraagt 1,5% en er dient maandelijks vooraf te worden afgelost.

We bepalen eerst het werkelijk geleende bedrag:

provisie:  $150.000 \cdot 0.015 = 2250,-$

aflossing eerste maand:  $15.000/12 = 1250,-$

werkelijk geleend bedrag:  $150.000 - 2250 - 1250 = 146.500$

Aan het eind van het eerste jaar bedraagt de resthypotheek 147.000 gulden.

#### \*\*\*\* RENTABILITEIT \*\*\*\*

Investering : Fl 146500.00

periode	winst (Fl.)
1	1250.00
2	1250.00
3	1250.00
4	1250.00
5	1250.00
6	1250.00
7	1250.00
8	1250.00
9	1250.00
10	1250.00
11	1250.00
12	147000.00

Rentabiliteit : 0.81 %

Het werkelijk betaalde rentepercentage per maand blijkt nu 0,81%

Op jaarbasis is dat 10,1%, dus ruim 2% hoger!

Programma:

```
10 ' *** RENTABILITEIT ***
20 ' ** invoer gegevens **
30 GOSUB 90
40 ' ** berekening **
50 GOSUB 240
60 ' ** uitvoer **
70 GOSUB 370
80 END
90 ' ** invoer gegevens **
100 CLS:LOCATE 5,0
110 PRINT"**** RENTABILITEIT ****"
120 LOCATE 0,4
130 INPUT"Investering      F1 ";CW
140 INPUT"Aantal perioden    : ";N
150 PRINT
160 DIM W(N)
170 FOR T=1 TO N
180 PRINT USING"Winst in periode ## ";T;
190 INPUT": ";W(T)
200 NEXT T
210 PRINT
220 PRINT"Even geduld aub"
230 RETURN
240 ' ** berekening **
250 OG=0
260 BG=1
270 BCW=0
280 I=(OG+BG)/2
290 IF (BG-OG)<1E-05 THEN 350
300 FOR T=1 TO N
310 BCW=BCW+W(T)/(1+I)^T
320 NEXT T
330 IF BCW>=CW THEN OG=I ELSE BG=I
340 GOTO 270
350 R=I*100
360 RETURN
370 ' ** uitvoer **
380 CLS:LOCATE 4,0
390 PRINT"**** RENTABILITEIT ****"
400 LOCATE 0,4
410 PRINT USING"Investering      : F1 #####.##";C
W
420 PRINT
```

```
430 PRINT"periode      winst (Fl.)"
440 FOR T=1 TO N
450 PRINT USING "    ##";T;
460 PRINT USING "          #####.##";W(T)
470 NEXT T
480 PRINT
490 PRINT USING "Rentabiliteit  : #####.## %";R
500 PRINT
510 RETURN
```

## 4 ALGEMEEN

Dit hoofdstuk begint met een aantal programma's, welke met name voor elektrotechnici interessant zijn, namelijk een programma om verzwakkers te berekenen en twee programma's ter berekening van impedanties van karakteristieke netwerken.

Vervolgens komt een programma aan bod waarmee het mogelijk is op eenvoudige manier „songs" te genereren. Dit programma gaat op een zeer originele wijze om met eenvoudige invoer.

We vervolgen met nog een programma dat de creativiteit stimuleert: een „tekenprogramma". Hiermee is het mogelijk fraaie tekeningen te maken.

Een schaakklok vormt het volgende onderwerp van dit hoofdstuk. Professioneel schaken is nu dus mogelijk geworden.

Dan gaan we over tot een programma dat eveneens met creativiteit te maken heeft, al is dit programma wel meer specifiek geschikt voor programmeurs. Het is een programma waarmee het mogelijk is de loop van een programma te traceren en de gebruikte variabelen op te sporen. Kortom, een onmisbare programmeerhulp.

We sluiten af met een checksumgenerator. Dit is een programma waarmee controlegetallen toegevoegd kunnen worden aan een programma. Dit om correct intypen te verzekeren.

### 4.1.1 VERZWAKKERS

Dit programma berekent de weerstandswaarden voor twee typen verzwakkers, namelijk een Pi- en een T-verzwakker, bij een opgegeven verzwakking.

De gebruiker kiest de impedanties (in Ohm) aan weerszijden van het netwerk. De computer berekent vervolgens de minimale verzwakking die bereikt kan worden en vraagt dan naar de gewenste verzwakking (in dB).

De computer tekent de verzwakkers, berekent de vereiste weerstandswaarden en drukt vervolgens de karakteristieke impedantie, de verzwakking en de weerstandswaarden af.

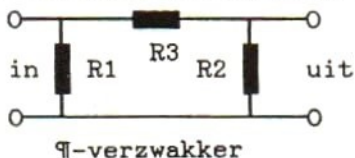
*Voorbeeld:*

```
**** VERZWAKKER NETWERKEN ****
```

```
Impedantie Z1 ( $\Omega$ ) ? 50
Impedantie Z2 ( $\Omega$ ) ? 60
Minimale demping      : 3.7654 dB
Gewenste demping (dB) ? 6
```



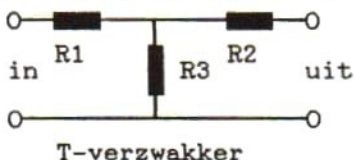
\*\*\*\* VERZWAKKER NETWERKEN \*\*\*\*



Impedantie Z1 = 50 Ω  
Impedantie Z2 = 60 Ω  
Demping = 6 dB

R1 = 111.38 Ω  
R2 = 293.378 Ω  
R3 = 40.9169 Ω

\*\*\*\* VERZWAKKER NETWERKEN \*\*\*\*



Impedantie Z1 = 50 Ω  
Impedantie Z2 = 60 Ω  
Demping = 6 dB

R1 = 10.2257 Ω  
R2 = 26.9347 Ω  
R3 = 73.3193 Ω

Programma:

```
10 ' *** VERZWAKKER NETWERKEN ***
20 CLS:WIDTH 39:KEY OFF
30 PRINT" **** VERZWAKKER NETWERKEN ****"
40 LOCATE 0,4
50 ' ** invoer **
60 GOSUB 270
70 ' ** berekenen en uitvoer **
80 CLS
90 PRINT" **** VERZWAKKER NETWERKEN ****"
100 IF A$="T" OR A$="t" THEN GOSUB 130 ELSE GOSUB
  200
110 LOCATE 0,22
120 END
```

```

130 ' ** T-verzwakker **
140 R3=ZS*2*SQR(P)/(1-P)
150 R1=Z1*X-R3
160 R2=Z2*X-R3
170 GOSUB 440' print gegevens
180 GOSUB 510' verzwakker tekenen
190 RETURN
200 ' ** Π-verzwakker **
210 R3=ZS*(1-P)/(2*SQR(P))
220 R1=1/(X/Z1-1/R3)
230 R2=1/(X/Z2-1/R3)
240 GOSUB 440' print gegevens
250 GOSUB 620' verzwakker tekenen
260 RETURN
270 ' ** invoer **
280 INPUT"Impedantie Z1 (Ω) ";Z1
290 INPUT"Impedantie Z2 (Ω) ";Z2
300 V=Z1/Z2
310 IF V<1 THEN V=1/V
320 E=20*LOG(SQR(V)+SQR(V-1))/LOG(10)
330 PRINT"Minimale demping      : "CSNG(E)"dB"
340 INPUT"Gewenste demping (dB) ";D
350 IF D<E THEN 330
360 P=10^(-D/10)
370 X=(1+P)/(1-P)
380 ZS=SQR(Z1*Z2)
390 PRINT"Π- of T-verzwakker (P/T) "
400 A$=INKEY$:IF A$="" THEN 400
410 IF A$="P" OR A$="T" OR A$="p" OR A$="t" THEN
RETURN
420 PRINT"Verkeerde invoer !":GOTO 390
430 RETURN
440 ' * uitvoer berekeningen *
450 LOCATE 0,11:SP$="      "
460 GOSUB 730:PRINT
470 PRINT SP$;"R1 ="CSNG(R1)"Ω"
480 PRINT SP$;"R2 ="CSNG(R2)"Ω"
490 PRINT SP$;"R3 ="CSNG(R3)"Ω"
500 RETURN
510 ' ** teken T-verzwakker **
520 LOCATE 0,2
530 PRINT SP$"O ————|————— O"
540 PRINT SP$"          |"
550 PRINT SP$"          ■      R2"
560 PRINT SP$"          ■ R3      uit"
570 PRINT SP$"          ■"

```

```

580 PRINT SP$"          |"
590 PRINT SP$"O-----O"
600 PRINT:PRINT SP$;"    T-verzwakker"
610 RETURN
620 ' * teken ¶-verzwakker *
630 LOCATE 0,2
640 PRINT SP$"O-----O"
650 PRINT SP$"          |"
660 PRINT SP$"          R3          |"
670 PRINT SP$"in ■ R1          R2 ■ uit"
680 PRINT SP$"          |          |"
690 PRINT SP$"          |          |"
700 PRINT SP$"O-----O"
710 PRINT:PRINT SP$;"    ¶-verzwakker"
720 RETURN
730 ' * print ingevoerde gegevens *
740 PRINT:PRINT
750 PRINT SP$;"Impedantie Z1 =";Z1"Ω"
760 PRINT SP$;"Impedantie Z2 =";Z2"Ω"
770 PRINT SP$;"Demping          ="D"dB"
780 RETURN

```

#### 4.1.2 SERIEEL NETWERK

In de voorbeelden zien we het seriële netwerk waar het hier om gaat. Weerstand R2, condensator C en spoel L staan in serie, terwijl parallel aan deze serieschakeling weerstand R1 is geschakeld.

Na het invoeren van de waarden van R1, R2, C en L krijgt de gebruiker een menu te zien. Dit menu biedt hem de volgende mogelijkheden:

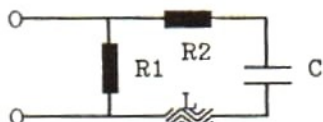
1. Verandering van de waarden van R1, R2, C en L.
2. Bepaling van de resonantiefrequentie en de impedantie bij deze frequentie.
3. Bepaling van de impedantie bij een gegeven frequentie.
4. Einde programma.

Van de impedantie wordt de modulus en het argument (in graden) afgedrukt.

Na het uitvoeren van de keuzes 1, 2 of 3 en het tonen van de resultaten, keert het programma door het indrukken van een willekeurige toets terug naar het menu.

Voorbeeld:

\*\*\*\* SERIEKRING \*\*\*\*



R1 ( $\Omega$ ) ? 2500  
R2 ( $\Omega$ ) ? 1000  
C ( $\mu\text{F}$ ) ? .1  
L ( $\mu\text{H}$ ) ? .01E+6

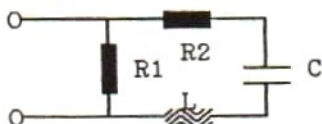
\*\*\*\* SERIEKRING \*\*\*\*

Mogelijkheden:

- 1 : veranderen componentwaarden
- 2 : bepalen resonantiefrequentie
- 3 : bepalen impedantie
- 4 : einde programma

Keuze ? 2

\*\*\*\* SERIEKRING \*\*\*\*



R1 = 2500  $\Omega$       C = .1  $\mu\text{F}$   
R2 = 1000  $\Omega$       L = 10000  $\mu\text{H}$

Resonantiefrequentie = 5032.92 Hz

Impedantie:mod = 714.286  $\Omega$   
arg = 0 graden

Druk een toets in voor menu..

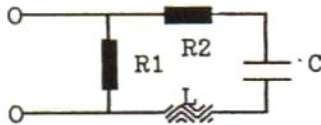
Mogelijkheden:

- 1 : veranderen componentwaarden
- 2 : bepalen resonantiefrequentie
- 3 : bepalen impedantie
- 4 : einde programma

Keuze ? 3

Bij welke frequentie ? 3000

\*\*\*\* SERIEKRING \*\*\*\*



R1 = 2500  $\Omega$       C = .1  $\mu$ F  
 R2 = 1000  $\Omega$       L = 10000  $\mu$ H  
 F = 3000 Hz

Impedantie:mod = 751.33  $\Omega$   
 arg = -13.301 graden

Druk een toets in voor menu..

Programma:

```

10 ' *** SERIEKRING ***
20 ' ** invoer **
30 GOSUB 570 **
40 ' ** menu **
50 GOSUB 740
60 ON K GOSUB 570,630,680
70 IF K<>4 THEN 40
80 END
90 ' * kop *
100 CLS:LOCATE 5,0
110 PRINT"**** SERIEKRING ****"
120 RETURN
130 ' * teken schakeling *
140 LOCATE 0,3:SP$=" "
150 PRINT SP$" O-----"
160 PRINT SP$;SP$" |"
170 PRINT SP$;SP$"■ R2 |"
180 PRINT SP$;SP$"■ R1 --- C"
190 PRINT SP$;SP$"■"
200 PRINT SP$;SP$" | L |"
210 PRINT SP$" O-----"
220 RETURN
230 ' ** invoer komponentwaarden **
240 ' ** invoer komponentwaarden **
250 PRINT
260 INPUT"R1 ( $\Omega$ ) ";R1
270 INPUT"R2 ( $\Omega$ ) ";R2
    
```

```

280 INPUT"C (μF) ";C:C=C/1000000!
290 INPUT"L (μH) ";L:L=L/1000000!
300 RETURN
310 ' ** bereken impedantie **
320 HULP=2*PI*F*L-1/(2*PI*F*C)
330 IM=HULP*R1^2
340 RE=R1^2*R2+R1*R2^2+R1*HULP^2
350 ZN=(R1+R2)^2+HULP^2
360 ZA=SQR(RE^2+IM^2)/ZN
370 ARG=180*ATN(IM/RE)/PI' in graden
380 RETURN
390 ' ** uitvoer **
400 GOSUB 90' * kop *
410 GOSUB 130' * teken schakeling *
420 PRINT:PRINT
430 PRINT"R1 ="CSNG(R1)"Ω";
440 PRINTTAB(17)"C ="CSNG(C*1000000!)"μF"
450 PRINT"R2 ="CSNG(R2)"Ω";
460 PRINTTAB(17)"L ="CSNG(L*1000000!)"μH"
470 PRINT
480 IF K=2 THEN PRINT"Resonantiefrequentie ";ELSE
PRINT"F ";
490 PRINT "="CSNG(F)"Hz"
500 PRINT
510 PRINT "Impedantie:mod ="CSNG(ZA)"Ω"
520 PRINT SP$" arg ="INT(ARG*1000+.5)/1000"gra
den
530 PRINT:PRINT
540 PRINT"Druk een toets in voor menu.."
550 IF INKEY$="" THEN 550
560 RETURN
570 ' ** invoer **
580 PI=3.141593#
590 GOSUB 90' * kop *
600 GOSUB 130' * teken schakeling *
610 GOSUB 240' * invoer waarden *
620 RETURN
630 ' ** resonantiefrequentie **
640 F=1/(2*PI*SQR(L*C))
650 GOSUB 310' bereken impedantie
660 GOSUB 390' uitvoer
670 RETURN
680 ' ** bepaling impedantie **
690 PRINT
700 INPUT"Bij welke frequentie ";F
710 GOSUB 310' bereken impedantie

```

```

720 GOSUB 390' uitvoer
730 RETURN
740 ' ** menu **
750 GOSUB 90' kop
760 LOCATE 0,4
770 PRINT"Mogelijkheden:"
780 PRINT
790 PRINT"1 : veranderen componentwaarden"
800 PRINT"2 : bepalen resonantiefrequentie"
810 PRINT"3 : bepalen impedantie"
820 PRINT"4 : einde programma"
830 PRINT
840 INPUT"Keuze ";K
850 IF K<1 OR K>4 THEN PRINT"Verkeerde invoer:got
o 890
860 RETURN

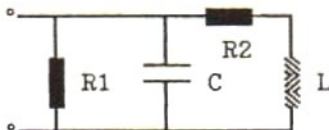
```

#### 4.1.3 PARALLEL NETWORK

In dit geval gaat het om een netwerk, dat een parallelschakeling is van weerstand R1, condensator C en de serieschakeling van weerstand R2 en spoel L. Zie hiervoor het voorbeeld. Het programma dient op de zelfde manier te worden gebruikt als dat voor het seriële netwerk.

Voorbeeld:

\*\*\*\* PARALLELKRING \*\*\*\*



```

R1 (Ω) ? 1000
R2 (Ω) ? 100
C (μF) ? .1
L (μH) ? .01E+6

```

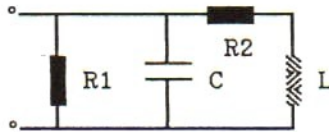
\*\*\*\* PARALLELKRING \*\*\*\*

Mogelijkheden:

- 1 : veranderen componentwaarden
- 2 : bepalen resonantiefrequentie
- 3 : bepalen impedantie
- 4 : einde programma

Keuze ? 2

\*\*\*\* PARALLELKRING \*\*\*\*



$$\begin{array}{ll} R1 = 1000 \Omega & C = .1 \mu\text{F} \\ R2 = 100 \Omega & L = 10000 \mu\text{H} \end{array}$$

Resonantiefrequentie = 4774.65 Hz

Impedantie:mod = 500  $\Omega$   
arg = 0 graden

Druk een toets in voor menu..

\*\*\*\* PARALLELKRING \*\*\*\*

Mogelijkheden:

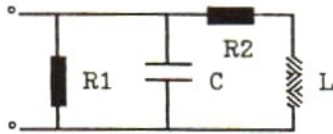
- 1 : veranderen componentwaarden
- 2 : bepalen resonantiefrequentie
- 3 : bepalen impedantie
- 4 : einde programma

Keuze ? 3

Bij welke frequentie ? 3000



\*\*\*\* PARALLELKRING \*\*\*\*



R1 = 1000 Ω      C = .1 μF  
 R2 = 100 Ω      L = 10000 μH

F = 3000 Hz

Impedantie:mod = 255.641 Ω  
 arg = 35.203 graden

Druk een toets in voor menu..

Programma:

```

10 ' *** PARALLELKRING ***
20 ' ** invoer **
30 GOSUB 580 **
40 ' ** menu **
50 GOSUB 760
60 ON K GOSUB 580,640,700
70 IF K<>4 THEN 40
80 END
90 ' * kop *
100 CLS:LOCATE 5,0
110 PRINT"**** PARALLELKRING ****"
120 RETURN
130 ' * teken schakeling *
140 LOCATE 0,3:SP$=" "
150 PRINT SP$" °-----"
160 PRINT SP$" | | | |"
170 PRINT SP$" | | R2 |"
180 PRINT SP$" | | | |"
190 PRINT SP$" | R1 | C | L"
200 PRINT SP$" | | | |"
210 PRINT SP$" | | | |"
220 PRINT SP$" °-----"
230 RETURN
240 ' ** invoer komponentwaarden **
250 PRINT
260 INPUT"R1 (Ω) ";R1
    
```

```

270 INPUT "R2 (Ω) ";R2
280 INPUT "C (μF) ";C:C=C/1000000!
290 INPUT "L (μH) ";L:L=L/1000000!
300 RETURN
310 ' ** bereken impedantie **
320 W=2*PI*F
330 N=R2^2+(W*L)^2
340 RE=1/R1+R2/N
350 IM=W*(C-L/N)
360 MDL=1/SQR(RE^2+IM^2)
370 IF RE=0 THEN ARG=-90*SGN(IM) ELSE ARG=-180*AT
N(IM/RE)/PI
380 RETURN
390 ' ** uitvoer **
400 GOSUB 90' * kop *
410 GOSUB 130' * teken schakeling *
420 PRINT:PRINT
430 PRINT "R1 ="CSNG(R1)"Ω";
440 C1=C*1000000!
450 PRINTTAB(17)"C ="CSNG(C*1000000!)"μF"
460 PRINT "R2 ="CSNG(R2)"Ω";
470 PRINTTAB(17)"L ="CSNG(L*1000000!)"μH"
480 PRINT
490 IF K=2 THEN PRINT "Resonantiefrequentie ";ELSE
PRINT "F ";
500 PRINT "="CSNG(F)"Hz"
510 PRINT
520 PRINT "Impedantie:mod ="CSNG(MDL)"Ω"
530 PRINT SP$" arg ="INT(ARG*1000+.5)/1000"gra
den
540 PRINT:PRINT
550 PRINT "Druk een toets in voor menu.."
560 IF INKEY$="" THEN 560
570 RETURN
580 ' ** invoer **
590 PI=3.141593#
600 GOSUB 90' * kop *
610 GOSUB 130' * teken schakeling *
620 GOSUB 240' * invoer waarden *
630 RETURN
640 ' ** resonantiefrequentie **
650 W2=1/(L*C)-R2^2/L^2
660 IF W2>0 THEN F=SQR(W2)/(2*PI) ELSE F=0
670 GOSUB 310' bereken impedantie
680 GOSUB 390' uitvoer
690 RETURN

```

```

700 ' ** bepaling impedantie **
710 PRINT
720 INPUT"Bij welke frequentie ";F
730 GOSUB 310' bereken impedantie
740 GOSUB 390' uitvoer
750 RETURN
760 ' ** menu **
770 GOSUB 90' kop
780 LOCATE 0,4
790 PRINT"Mogelijkheden:"
800 PRINT
810 PRINT"1 : veranderen komponentwaarden"
820 PRINT"2 : bepalen resonantiefrequentie"
830 PRINT"3 : bepalen impedantie"
840 PRINT"4 : einde programma"
850 PRINT
860 INPUT"Keuze ";K
870 IF K<1 OR K>4 THEN PRINT"Verkeerde invoer:got
o 890
880 RETURN

```

## 4.2 SONGGENERATOR

Iedereen die wel eens geprobeerd heeft een muziekstukje op de MSX-computer te componeren en in te voeren zal gemerkt hebben dat dit, ondanks het speciaal hiervoor ontwikkelde MusicMacro, een tijdrovend karwei is.

SONGGENERATOR biedt hiervoor een snelwerkend alternatief.

Uitgaand van de door u vastgelegde grote lijn, die u in de standaard accoord-notatie kunt invoeren, zoekt SONGGENERATOR bijpassende noten, die vervolgens door u gewijzigd en afgespeeld kunnen worden.

Een van de aantrekkelijke kanten hierbij is dat SONGGENERATOR een toevalskarakter heeft, zodat niet steeds dezelfde tonen tevoorschijn komen.

Achtereenvolgens worden nu de songstructuur en de commando's van SONGGENERATOR beschreven.

Daarna zal een voorbeeld worden behandeld.

### SONGGENERATOR: de songstructuur

SONGGENERATOR beschouwd een compositie als verdeeld in MATEN, PARTS en SONGS. Een PART is hierbij niets anders dan een serie MATEN en evenzo is een SONG een serie PARTS.

Het eerste dat dient te gebeuren is de invoer van een of meer MATEN.  
U typt als eerste commando : -

>EM (gevolgd door RETURN)

(> is de zogenaamde prompt)

Nu verschijnt een vraagteken als prompt en de mededeling E MAAT, ten teken dat u zich in de edit-mode voor maten bevindt.

Nu kunt u een MAAT invoeren. Deze dient een nummer of letter als naam te krijgen. Een voorbeeld :

s = dmdmdm7dm (gevolgd door RETURN)

Deze MAAT van vier accoorden, namelijk drie dm (D mineur) accoorden en één dm7 (D mineur septiem) accoord, zal nu op het scherm geplaatst zijn. De d's in de accoordstring zullen hierbij in hoofdletters afgedrukt worden.

In de string kunnen de volgende typen accoorden staan:

D	majeur accoord	D#	Dis accoord
Dm	mineur accoord	Ds	Dsus accoord
D-	Des accoord	D7	D septiem accoord

In plaats van het D accoord kunt u natuurlijk ook een C,E,F,G,A of B accoord nemen. Combinaties hiervan zijn ook mogelijk bijvoorbeeld: Em7#.

Wanneer het geluid van de TV voldoende sterk staat afgesteld of (beter) de geluidoutput van uw MSX-computer op een versterker aangesloten is, kunt u horen welke noten SONG-GEN gekozen heeft.

U kunt tot maximaal 8 MATEN invoeren.

De edit-mode wordt verlaten door alleen een RETURN te geven.

Nu moet er een PART samengesteld worden. We toetsen in:

>EP (gevolgd door RETURN)

We kunnen de verschillende PARTS invoeren bijvoorbeeld:

i = abaCc3abc (gevolgd door RETURN)

waarbij dan a,b,C,c en 3 namen van door u gemaakte MATEN zijn.

Hier geldt ook weer een maximum van 8 PARTS.

Na het opnieuw verlaten van de edit-mode (door het geven van alleen RETURN) kunnen we de SONGS gaan invoeren :

>ES (gevolgd door RETURN)

De invoer geschiedt analoog aan die van PARTS, hier geldt echter een maximum van 5 SONGS.

## SONGGENERATOR : de commando's en parameters

Daar de parameters in de commando-mode gewijzigd moeten worden, worden ze hier ook beschouwd.

CSAVE f\$	schrijf de datafiles en parameters weg naar disk of cassette onder de naam f\$
CLOAD f\$	lees datafiles en parameters in onder de naam f\$
MUSIC f\$	Voer MusicMacro f\$ uit. Met dit commando stellen we de klankkleur en het tempo in. Zie de PLAY-instructie in uw MSX-handleiding voor details hierover
RERAND f\$	Randomiseer de MATEN f\$ opnieuw bv :RE abc (gevolgd door RETURN)
EM,EP,ES	ga naar edit-mode van resp. MAAT, PART of SONG
PM,PP,PS f\$	speel de namen in f\$ uit resp. MAAT, PART of SONG
SPEED n	wijzig de speelsnelheid
NOTEN n	wijzig het aantal herhalingen van ieder accoord
VOICES n	wijzig het aantal stemmen (maximaal 3)
WIDTH n	stelt het toonbereik in waarbinnen SONGGEN noten kiest
SILENCE n	stelt de noot/rust verhouding in. Als n=0 worden er alleen noten genereerd, als n=9 vrijwel alleen rusten.
HELP	druk een samenvatting van de commando's af
\	herhaal het laatste commando. Dit is handig bij het randomiseren: Na het commando RE f\$ toetst men, totdat het resultaat naar tevredenheid is, \' (gevolgd door RETURN) in. Dit teken heet „backslash”.
EXIT	verlaat SONGGEN

Deze commando's werken alleen in de commando-mode, dus niet in de edit-mode. Alle commando's mogen worden afgekort tot de eerste twee letters.

Verder kan met de F1-toets het spelen van een compositie afgebroken worden. Dit is belangrijk omdat de composities al gauw een tijdsduur van ettelijke minuten kunnen beslaan.

### Een voorbeeld

Het volgende voorbeeld beschrijft de invoer van enkele MATEN, PARTS en SONGS. De compositie is vrij willekeurig gekozen.

Aangenomen dat de computer in de commando-mode staat toetsen we in:

EM (gevolgd door RETURN)

en voeren vervolgens de volgende regels in:

a = amamam	g = gsgsgs
e = ememem	c = cmcscms
A = amam7am7	D = dm7dm7dm7dm7
d = amamdman	

en vervolgens alleen een RETURN om weer in de commando-mode te komen.  
Geef nu:

EP (gevolgd door RETURN)

en voer in:

```
i = aeae          l = gccgcc
j = AeAe          o = D
h = dDdD
```

Geef weer alleen een RETURN en vervolgens:


```
ES          (gevolgd door RETURN)
```

voer in:

```
x = ijijji
y = hh
z = olloll
```

en nogmaals een RETURN.

**Resultaat:**

```
> 
-----
MAAT a=AmAmAmAm          width: 8
     e=EmEmEmEm          voices: 3
     A=AmAmAm7Am7        silence: 6
     d=AmAmDmAm          noten: 1
     g=GsGsGs            speed: 5
     c=CmsCmsCms
     D=Dm7Dm7Dm7Dm7

PART i=aeae
     j=AeAe
     h=dDdD
     l=gccgcc
     o=D

SONG x=ijijijji
     y=hh
     z=olloll
```

Het is nu tijd om de parameters in te voeren en de MATEN te bepalen.

Geef:

```
SI 4
SP 8
MU m1500s0t200
WI 6
```

Speel met behulp van PM f\$ alle MATEN (a,e,A,d,g,c en D) af en maak ze zonodig opnieuw met RE f\$ totdat de MATEN naar wens zijn. Het gebruik van het herhaalcommando \ is hierbij handig.

Geef vervolgens:

```
WI 8
SI 6
```

Als alles goed gegaan is, is de compositie nu ingevoerd en kunt u hem saven met CS lied1 gevolgd door RETURN.

Met behulp van een afspeelcommando als:

```
PS xyyxyyz
```

wordt de hele compositie afgespeeld.

Veel succes bij het componeren.

*Programma:*

```
10 REM SONGGENERATOR
20 CLEAR 1000,&HCFE0:DEFINT H-N
30 KEY8,"csave"+CHR$(34)+"PLAY"+CHR$(34)+",2"+CHR
$(13)
40 KEY OFF:I=RND(-TIME)
50 POKE &HFBB0,1
60 ON STOP GOSUB 70:STOP ON:GOTO 80
70 RETURN
80 COLOR 1,15:SCREEN0,,0
90 CLS:WIDTH 39
100 GOSUB 4030'lees routine in
110 '-----
120 ' constanten
130 NX=15'rustlengte
140 NS=3'aantal stemmen
150 NB=1'nootrepetitie
160 ML=20'maatlengte
170 MA=8'aantal maten,delen,stukken
180 HM=0:HP=0:HS=0:LV=24:HA=0
190 HT=5
200 A$="m3000t150s0115"
210 PLAY A$,A$,A$
220 DIM NP(ML,MA,NS)
230 DIM NM$(MA),NP$(MA),NS$(MA)
240 DIM LM(MA),LT(7)
250 DIM L$(MA),LP$(MA),LS$(MA)
260 GOSUB 3600' vul lt(7)
270 ON KEY GOSUB 1380'play/stop
```

```

280 KEY(1) ON
290 ON ERROR GOTO 3720
300 '-----
310 ' MAIN
320 GOSUB 1750' print M P S
330 GOSUB 630' cmdint
340 COLOR 15, 4, 4:CLS:END
350 '-----
360 ' space killer
370 J=LEN(A$)
380 IF J=0 THEN RETURN
390 IF MID$(A$,J,1)=" " THEN J=J-1:GOTO 390
400 A$=LEFT$(A$,J)
410 RETURN
420 '-----
430 ' [a..g] >> [A..G]
440 LL=LEN(NM$(I))
450 FOR J=1 TO LL
460 B=ASC(MID$(NM$(I),J))
470 IF B>96 AND B<104 THEN NM$(I)=LEFT$(NM$(I),J-
1)+CHR$(B-32)+RIGHT$(NM$(I),LL-J)
480 NEXT J
490 RETURN
500 '-----
510 ' lc >> UC
520 LL=LEN(K$)
530 FOR I=1 TO LL
540 B=ASC(MID$(K$,I))
550 IF B>96 AND B<123 THEN K$=LEFT$(K$,I-1)+CHR$(
B-32)+RIGHT$(K$,LL-I)
560 NEXT I
570 RETURN
580 '-----
590 LOCATE 0,0:PRINTSPC(32)
600 LOCATE 0,0
610 RETURN
620 '-----
630 ' cmd int
640 LOCATE 0,1:PRINTSTRING$(39,"-")
650 J=LV/3:GOSUB 2060' width
660 J=NS:GOSUB 2000' aantal voices
670 J=HA:GOSUB 1560' silence
680 J=NB:GOSUB 1620'nootrep.
690 J=HT:GOSUB 1670'speed
700 GOSUB 590
710 LINEINPUT">";A$

```



```

720 IX=0
730 IF LEFT$(A$,1)="\" THEN A$=VO$
740 VO$=A$
750 LA=LEN(A$)
760 IF LA=1 THEN 980
770 K$=LEFT$(A$,2):GOSUB 520
780 IF LA=2 THEN 930
790 A$=RIGHT$(A$,LA-INSTR(3,A$," "))
800 J=VAL(A$)
810 IF K$="PM" THEN GOSUB 1420:GOTO 640 ' play ma
at
820 IF K$="PP" THEN GOSUB 1010:GOTO 640 ' play pa
rt
830 IF K$="PS" THEN GOSUB 1140:GOTO 640 ' play so
ng
840 IF K$="SI" THEN GOSUB 1550:GOTO 640 ' rusten
850 IF K$="WI" THEN GOSUB 2050:GOTO 640 ' aantal
oktaven
860 IF K$="SP" THEN GOSUB 1670:GOTO 640
870 IF K$="NO" THEN GOSUB 1620:GOTO 640 ' aant no
ten
880 IF K$="VO" THEN GOSUB 1990:GOTO 640 ' aantal
voices
890 IF K$="RE" THEN GOSUB 2140:GOTO 640 ' opnieuw
randomizen
900 IF K$="CL" THEN GOSUB 2310:GOSUB 1750:GOTO 64
0
910 IF K$="CS" THEN GOSUB 2570:GOTO 640
920 IF K$="MU" THEN GOSUB 2810:GOTO 640
930 IF K$="EM" THEN KE=1:GOSUB 2880:GOTO 640
940 IF K$="EP" THEN KE=2:GOSUB 2880:GOTO 640
950 IF K$="ES" THEN KE=3:GOSUB 2880:GOTO 640
960 IF K$="HE" THEN GOSUB 3780:GOTO 640
970 IF K$="EX" THEN RETURN
980 GOTO 640
990 '-----
1000 ' play part(s)
1010 P$=A$
1020 FOR IP=1 TO LEN(P$)
1030 B$=MID$(P$,IP,1)
1040 FOR JP=1 TO HP
1050 IF LP$(JP)=B$ THEN 1070
1060 NEXT JP
1070 A$=NP$(JP)
1080 IF IX=1 THEN 1110
1090 GOSUB 1440' speel maat

```

```

1100 NEXT IP
1110 RETURN
1120 '-----
1130 ' play song(s)
1140 Q$=A$
1150 FOR IS=1 TO LEN(Q$)
1160 B$=MID$(Q$, IS, 1)
1170 FOR JS=1 TO HS
1180 IF LS$(JS)=B$ THEN 1200
1190 NEXT JS
1200 P$=NS$(JS)
1210 IF IX=1 THEN 1240
1220 GOSUB 1020' speel part
1230 NEXT IS
1240 RETURN
1250 '-----
1260 ' note playing
1270 H1=NP(JJ, J, 1)
1280 H2=NP(JJ, J, 2)
1290 H3=NP(JJ, J, 3)
1300 FOR HB=1 TO NB
1310 IF H1=0 AND H2=0 AND H3=0 THEN PLAY "r=nx;",
"r=nx;", "r=nx;":GOTO 1350
1320 IF NS=3 THEN PLAY "n=h1;", "n=h2;", "n=h3;":GO
TO 1350
1330 IF NS=2 THEN PLAY "n=h1;", "n=h2;":GOTO 1350
1340 PLAY "n=h1;"
1350 NEXT HB
1360 RETURN
1370 '-----
1380 ' key(1)
1390 IX=1:GOSUB 590:PRINT"STOP
1400 RETURN
1410 '-----
1420 ' play maten
1430 'l$ bevat strings uit a$
1440 FOR I=1 TO LEN(A$)
1450 B$=MID$(A$, I, 1)
1460 FOR J=1 TO HM
1470 IF L$(J)=B$ THEN 1490
1480 NEXT J
1490 FOR JJ=1 TO LM(J)
1500 GOSUB 1260' play note
1510 NEXT JJ
1520 NEXT I
1530 RETURN

```

```

1540 '-----
1550 ' rusten inbrengen
1560 IF J<0 OR J>9 THEN 3650
1570 HA=J:LOCATE 28,5
1580 PRINT"silence:"HA
1590 RETURN
1600 '-----
1610 ' nootrepetitie
1620 IF J<1 OR J>8 THEN 3650
1630 NB=J:LOCATE 30,6
1640 PRINT"noten:"NB
1650 RETURN
1660 '-----
1670 ' speed
1680 IF J<1 OR J>23 THEN GOTO 3650
1690 LOCATE 30,7:PRINT"speed:";J
1700 HT=J:J=J*10+23
1710 PLAY "t=j;","t=j;","t=j;"
1720 RETURN
1730 '-----
1740 ' print M P S
1750 LOCATE 0,2:PRINT"MAAT"
1760 IF HM=0 THEN 1810
1770 FOR J=1 TO HM
1780 LOCATE 5,J+1:GOSUB 1960
1790 PRINTL$(J);"=";NM$(J)
1800 NEXT J
1810 LOCATE 0,2+MA:PRINT"PART"
1820 IF HP=0 THEN 1870
1830 FOR J=1 TO HP
1840 LOCATE 5,J+MA+1:GOSUB 1960
1850 PRINTLP$(J);"=";NP$(J)
1860 NEXT J
1870 LOCATE 0,2+2*MA:PRINT"SONG"
1880 IF HS=0 THEN 1930
1890 FOR J=1 TO HS
1900 LOCATE 5,J+2*MA+1:GOSUB 1960
1910 PRINTLS$(J);"=";NS$(J)
1920 NEXT J
1930 RETURN
1940 '-----
1950 ' space
1960 PRINTSPC(23):LOCATE 5,CSRLIN
1970 RETURN
1980 '-----
1990 ' aantal voices

```

```

2000 IF J<1 OR J>3 THEN 3650
2010 LOCATE 29,4:NS=J
2020 PRINT"voices:"NS
2030 RETURN
2040 '-----
2050 ' WI: aantal oktaven
2060 IF J<1 OR J>8 THEN 3650
2070 LOCATE 36,3
2080 PRINT" "
2090 LOCATE 30,3
2100 PRINT"width:"J
2110 LV=J*3
2120 RETURN
2130 '-----
2140 ' rerandomize
2150 FOR H=1 TO LEN(A$)
2160 B$=MID$(A$,H,1)
2170 FOR I=1 TO HM
2180 IF L$(I)=B$ THEN 2200
2190 NEXT I
2200 GOSUB 3270' randomize np(,i,)
2210 NEXT H
2220 RETURN
2230 '-----
2240 GOSUB 590:LOCATE 1,0:STOP OFF
2250 INPUT"disk of tape (d/t)";K$
2260 K$=LEFT$(K$,1):GOSUB 520
2270 IF K$="D" THEN K$="":RETURN
2280 IF K$="T" THEN K$="cas:":RETURN
2290 GOTO 2240
2300 '-----
2310 ' file laden
2320 GOSUB 2240
2330 OPEN K$+A$ FOR INPUT AS #1
2340 INPUT#1,NS,NB,HM,HP,HS,HA,LV
2350 FOR I=1 TO HM
2360 INPUT#1,NM$(I)
2370 INPUT#1,LM(I)
2380 INPUT#1,L$(I)
2390 NEXT I
2400 FOR I=1 TO HP
2410 INPUT#1,NP$(I)
2420 INPUT#1,LP$(I)
2430 NEXT I
2440 FOR I=1 TO HS
2450 INPUT#1,NS$(I)

```

```

2460 INPUT#1,LS$(I)
2470 NEXT I
2480 FOR I=1 TO HM
2490 FOR J=1 TO LM(I)
2500 FOR K=1 TO NS
2510 INPUT#1,NP(J,I,K)
2520 NEXT K:NEXT J:NEXT I
2530 CLOSE #1
2540 RETURN
2550 '-----
2560 ' file wegschrijven
2570 SCREEN ,,,1:GOSUB 2240
2580 OPEN K$+A$ FOR OUTPUT AS #1
2590 PRINT#1,NS,NB,HM,HP,HS,HA,LV
2600 FOR I=1 TO HM
2610 PRINT#1,NM$(I)
2620 PRINT#1,LM(I)
2630 PRINT#1,L$(I)
2640 NEXT I
2650 FOR I=1 TO HP
2660 PRINT#1,NP$(I)
2670 PRINT#1,LP$(I)
2680 NEXT I
2690 FOR I=1 TO HS
2700 PRINT#1,NS$(I)
2710 PRINT#1,LS$(I)
2720 NEXT I
2730 FOR I=1 TO HM
2740 FOR J=1 TO LM(I)
2750 FOR K=1 TO NS
2760 PRINT#1,NP(J,I,K)
2770 NEXT K:NEXT J:NEXT I
2780 CLOSE #1
2790 RETURN
2800 '-----
2810 ' MM routine
2820 K$=A$:GOSUB 520:I=INSTR(K$,"L")
2830 IF I<1 THEN 2850
2840 NX=VAL(RIGHT$(K$,LEN(K$)-I))
2850 PLAY K$,K$,K$
2860 RETURN
2870 '-----
2880 'edit M,P,S
2890 LOCATE 32,0:PRINT"E ";
2900 IF KE=1 THEN PRINT"MAAT"ELSE IF KE=2 THEN PR
INT"PART" ELSE PRINT"SONG"

```

```

2910 GOSUB 590
2920 INPUT A$:A$=LEFT$(A$,22)
2930 GOSUB 360' kill space
2940 IF INSTR(A$,"")=0 THEN 3140
2950 K$=LEFT$(A$,1)
2960 IF K$="" THEN 3140
2970 IF KE=1 THEN HX=HM ELSE IF KE=2 THEN HX=HP ELSE HX=HS
2980 IF HX=0 THEN I=1:HX=1:GOTO 3050
2990 J=1
3000 GOSUB 3160
3010 I=J
3020 IF KE=3 AND (I>(21-2*MA)) THEN 3140
3030 IF I>MA THEN 3650
3040 IF I>HX THEN HX=I
3050 IF KE=1 THEN L$(I)=K$:HM=HX ELSE IF KE=2 THEN LP$(I)=K$:HP=HX ELSE LS$(I)=K$:HS=HX
3060 K$=RIGHT$(A$,LEN(A$)-2)
3070 IF KE=1 THEN NM$(I)=K$ ELSE IF KE=2 THEN NP$(I)=K$ ELSE NS$(I)=K$
3080 IF K$="" THEN 3140
3090 IF KE=1 THEN GOSUB 430' LC>>UC
3100 GOSUB 1750' PRINT
3110 IF KE=1 THEN GOSUB 3270' randomize
3120 GOTO 2910
3130 ' space
3140 LOCATE 32,0:PRINTSPC(6)
3150 RETURN
3160 IF J>HX THEN RETURN
3170 ON KE GOTO 3180,3200,3220
3180 IF L$(J)=K$ THEN RETURN
3190 GOTO 3230
3200 IF LP$(J)=K$ THEN RETURN
3210 GOTO 3230
3220 IF LS$(J)=K$ THEN RETURN
3230 J=J+1:GOTO 3160
3240 '-----
3250 ' randomize np(,i,)
3260 ' nm$(i) bevat acc string
3270 B$=NM$(I):LL=LEN(B$)
3280 J=1:IN=0
3290 AC$=MID$(B$,J,1):C$=""
3300 J=J+1
3310 IF J>(LL+1) THEN 3350
3320 D$=MID$(B$,J,1)
3330 IA=INSTR("CDEFGAB",D$)

```

```

3340 IF IA=0 THEN C$=C$+D$:GOTO 3300
3350 IN=IN+1
3360 ID=-1:MI=0:MX=3
3370 IA=INSTR("CDEFGAB",AC$)
3380 IA=LT(IA)
3390 IF INSTR(C$,"7") THEN ID=0:MX=4
3400 IF INSTR(C$,"s") THEN MX=4
3410 IF INSTR(C$,"m") THEN MI=-1
3420 IF INSTR(C$,"#") THEN IA=IA+1
3430 IF INSTR(C$,"-") THEN IA=IA-1
3440 FOR JJ=1 TO NS
3450 NO=INT(RND(1)*LV)
3460 OK=(INT(NO/5)+1)*12
3470 NO=(NO MOD MX)*3+1
3480 IF (OK/3)MOD3=0 THEN NP(IN,I,JJ)=OK+IA:GOTO
3550
3490 IF NO=10 AND ID THEN NO=11
3500 IF NO=4 AND MI THEN NO=3
3510 IF NO=1 THEN NO=0
3520 NP(IN,I,JJ)=NO+IA+OK
3530 IR=RND(1)<(.11*HA)
3540 IF IR THEN NP(IN,I,JJ)=0
3550 NEXT JJ
3560 PLAY "n=NP(IN,I,1);","n=NP(IN,I,2);","n=np(i
n,i,3);"
3570 IF J=LL+1 THEN LM(I)=IN:RETURN
3580 GOTO 3290
3590 '-----
3600 ' vul lt(7)
3610 LT(1)=0:LT(2)=2:LT(3)=4:LT(4)=5
3620 LT(5)=7:LT(6)=9:LT(7)=11
3630 RETURN
3640 '-----
3650 ' foutjes
3660 LOCATE 25,0
3670 PRINT"fout commando
3680 FOR I=1 TO 2000 :NEXT I
3690 LOCATE 25,0:PRINTSPC(13)
3700 RETURN
3710 '-----
3720 STOP ON:IF ERR=5 THEN 3760
3730 PRINT"Error "ERR" probeer opnieuw "
3740 TIME=0
3750 IF TIME<200 THEN 3750
3760 RESUME 640
3770 '-----

```

```

3780 ' help file
3790 IF HY=1 THEN U=USR1(0):GOTO 4010
3800 CLS:LOCATE 6,0
3810 PRINT" SONGGEN COMMANDOLIJST
3820 PRINTSTRING$(38,"-")
3830 PRINT"  CLOAD f$TAB(14)":laad datafile f$
3840 PRINT"  CSAVE f$TAB(14)":bewaar datafile f$
3850 PRINT"  MUSIC f$TAB(14)":MusicMacro
3860 PRINT"  SPEED nTAB(14)":speelsnelheid
3870 PRINT"  NOTES nTAB(14)":notherhalingen
3880 PRINT"  WIDTH nTAB(14)":notenbereik
3890 PRINT"  VOICES nTAB(14)":aantal stemmen
3900 PRINT"SILENCE nTAB(14)":noot/rust verhoudin
g
3910 PRINT"  RERAND f$TAB(14)":maak maten f$ opni
euw
3920 PRINT"      \TAB(14)":herhaal commando
3930 PRINT"      E?TAB(14)":edit ? (? = M,P,S)
3940 PRINT"      P? f$TAB(14)":speel f$ uit ?
3950 PRINT"      HELPTAB(14)":print deze file
3960 PRINT"      EXITTAB(14)":verlaat PLAY
3970 LOCATE 0,18:PRINT"Een maat is bv:x=CmFCmF-s7
G#msCD
3980 PRINT"F1 onderbreekt spelen"
3990 LOCATE 24,22:PRINT"sla toets aan
4000 U=USR0(0):HY=1
4010 IF INKEY$="" THEN 4010
4020 CLS:GOSUB 1730:RETURN
4030 DEFUSR0=&HCFE0:DEFUSR1=&HCFE0
4040 FOR I=&HCFE0 TO &HCFE8
4050 READ A:POKE I,A:NEXT I
4060 DATA 33,0,0,17,0,224,1,192,3,195,89,0,201
4070 DATA 33,0,224,17,0,0,1,192,3,195,92,0,201
4080 RETURN

```



### 4.3 REMBRANDTS TOOLKIT

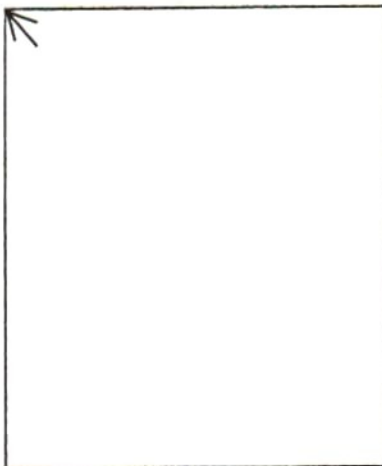
Zeker is dat Rembrandt nooit van een dergelijke toolkit gedroomd kan hebben. Hij zou ongetwijfeld zeer verbaasd zijn geweest als hij had kunnen zien hoe helemaal zonder penseel, potlood of ander tekenmateriaal toch echte tekeningen gemaakt kunnen worden. Voor ons is dat allemaal veel gewoner.

Dit programma biedt de mogelijkheid om met eenvoudige hulpmiddelen aardige plaatjes en schema's te maken. Essentieel natuurlijk is de opslag- en afdrukmogelijkheid van dit programma.

#### WERKING

Na het opstarten wordt gevraagd naar de naam van de tekening. Indien we een eerder gemaakte tekening willen wijzigen, geven we de naam van die tekening op. Zoniet, dan geven we een nieuwe naam op. Druk vervolgens op de RETURN-toets.

Nu verschijnt links op het scherm een roodgekleurd vierkant met in de linkerbovenhoek een pijl. Rechts naast het vierkant staat een overzicht van de functietoetsen en hun werking.



```
F1 : Draw  
F2 : Line  
F3 : Block  
F4 : Circle  
F5 : Paint  
F6 : Load  
F7 : Save  
F8 : Dump  
F9 : Clear  
F10: Quit
```

Als we gebruik maken van een gewone tv of een kleurenmonitor kunnen we zien dat het overzicht naast het vierkant een rode kleur heeft. Als we op een van de functietoetsen drukken dan verandert de kleur van de betreffende tekst in blauw. Hieraan kunnen we zien welke mode geactiveerd is. Als we met een monochroom monitor werken is het kleurverschil ook te zien, hoewel minder duidelijk.

Het is mogelijk om meer modes tegelijk te activeren.

We zullen nu de verschillende modes bespreken.

Zolang we nog op geen enkele functietoets gedrukt hebben is de zogenaamde dot-mode actief. Dit houdt in dat we met behulp van de joystick of cursortoetsen de pijl binnen het vierkant kunnen bewegen, zonder dat er iets verandert aan het vierkant. Er wordt dus niet getekend. Drukken we echter op de vuurknop of op de spatiebalk dan verschijnt op de plaats die door de pijl wordt aangewezen een punt op het scherm. Dot is het Engelse woord voor punt dus vandaar de naam dot-mode.

Door op **F1** te drukken verandert de dot-mode in de draw-mode. Rechts van het rode vierkant is de tekst **F1:draw** nu blauw gekleurd. In deze mode wordt de pijl overal gevolgd door een lijn. Ongeveer zoals we met een potlood, zonder de punt van het papier te nemen, kunnen tekenen. Willen we de lijn onderbreken dan moeten we nogmaals op **F1** drukken. Alle tekst rechts van het vierkant is dan weer rood en we zijn weer teruggekomen in de dot-mode.

Met **F2** activeren we de line-mode. In deze mode kunnen we lijnstukken tekenen. Dit gaat als volgt: op de plaats waar we een uiteinde van het lijnstuk willen hebben drukken we op de vuurknop of op de spatiebalk. Nu gaan we naar de plaats waar we het andere uiteinde van het lijnstuk willen hebben en drukken daar nogmaals op de vuurknop of spatiebalk. Onmiddellijk verschijnt het lijnstuk. Deze mode schakelen we weer uit door nogmaals op **F2** te drukken.

Dan komen we bij **F3** terecht. Dit is de Block-mode. Deze stelt ons in staat rechthoeken te tekenen. De manier van aangeven doet denken aan die van de line-mode. We moeten weer twee punten opgeven door op de spatiebalk of vuurknop te drukken. Deze twee punten worden dan gebruikt als twee schuin tegenover elkaar liggende hoekpunten van de rechthoek. De rechthoek wordt altijd opgebouwd uit horizontale en verticale lijnen. Het is dus zaak de twee punten niet op dezelfde hoogte te kiezen want dan is het resultaat een lijnstuk, een rechthoek die geen hoogte heeft.

Nadat we op de reeds vertrouwde manier **F3** weer uitgeschakeld hebben kiezen we **F4**, de circle-mode. Het is nu dus mogelijk cirkels te maken. Ook dit doen we weer door op de gebruikelijke wijze twee punten te kiezen. Het eerste punt bepaalt het middelpunt van de cirkel, het tweede is een punt ergens op de rand van de cirkel. Door de afstand tussen deze punten te variëren kunnen we de cirkelgrootte variëren.

Na deze mode weer uitgeschakeld te hebben komen we tenslotte bij de laatste van de echte tekenmodes, **F5** de paint-mode. Hiermee kunnen we dus „verven“. Hiertoe zetten we de pijlpunt in een gesloten vlak en drukken op de vuurknop of spatiebalk. Het vlak wordt nu wit. Op de printer zal dit later zwart zijn. Is het vlak niet goed gesloten dan zal het hele teken-scherm wit worden, voorzover tenminste geen gesloten vlakken daarin aanwezig zijn, deze blijven rood.

Alvorens we overgaan tot de bespreking van de overige functietoetsen willen we eerst nog een ander onderwerp behandelen; het maken van combinaties.

Het is mogelijk elke functie met een of meer andere te combineren. Als we bijvoorbeeld **F2** en **F4** activeren, ze zijn dan allebei blauw, dan krijgen we altijd cirkels met lijnstukken tegelijk. In dit geval wordt dus de straal van de cirkel in de cirkel getekend. Activeren we dan ook nog **F3** dan wordt tegelijkertijd een rechthoek (met een diagonaal) door de cirkel getekend.

De combinatie van **F4** en **F5** levert altijd ingekleurde cirkels op. Zo zijn er natuurlijk veel meer combinaties te maken, we laten het aan u over te bekijken welke leuk en zinvol zijn en welke niet.

De functietoetsen **F6** tot en met **F10** kunnen vanzelfsprekend niet in combinatie gebruikt worden.

Met **F6** kunnen we een vorige tekening laden en er verder aan werken, of deze afdrukken, net wat we willen. Als we moeilijke schema's of tekeningen willen maken is het aan te bevelen om vaak te saven (**F7**). Als er dan iets mis gaat met het tekenen hebben we altijd nog een vorige versie om vanuit te gaan. Met **F6** kunnen we die dan opnieuw laden.

De filenaam wordt bij het opstarten gevraagd en kan tussentijds niet gewijzigd worden. Met **F7** kan een file gesaved (opgeslagen) worden onder de bij het opstarten gegeven filenaam.

**F8** zorgt voor een zogenaamde screendump; een afdruk van het scherm op de MSX-printer, bijvoorbeeld de EPSON GX-80.

**F9** is een gevaarlijke opdracht. Clear veegt het scherm schoon. Een eventuele tekening is dus verdwenen als deze tenminste niet gesaved was.

**F10** tenslotte is al even gevaarlijk als F9. Quit betekent het scherm wissen en het programma verlaten. Als de tekening dus niet gesaved is is deze nu verdwenen.

We besluiten met een kort overzicht van de functietoetsen en hun werking:

- F1 - Draw** De pijl wordt overal gevolgd door een lijn.
- F2 - Line** Tussen twee aangegeven punten wordt een lijnstuk getrokken.
- F3 - Block** Twee aangegeven punten dienen als hoekpunten van een rechthoek.
- F4 - Circle** Twee aangegeven punten geven de straal aan van de gewenste cirkel.
- F5 - Paint** Het vlak waarin de pijlpunt staat wordt gevuld door op de vuurknop of spatiebalk te drukken.
- F6 - Load** Een eerder gemaakte tekening wordt geladen. De filenaam hebben we bij het opstarten van het programma al opgegeven.
- F7 - Save** De tekening wordt opgeslagen onder de bij het opstarten opgegeven naam.
- F8 - Dump** Het scherm wordt afgedrukt op een MSX-printer.
- F9 - Clear** Het scherm wordt gewist en de tekening is dus weg.
- F10 - Quit** Het scherm wordt gewist en het programma verlaten.

*Opmerkingen:*

Elke mode wordt weer uitgeschakeld door nogmaals op de betreffende functietoets te drukken.

Is geen enkele mode zichtbaar geactiveerd dan is de dot-mode actief.

De pijlpunt kan bestuurd worden door middel van een joystick en de vuurknop of door de cursortoetsen en de spatiebalk.

Programma:

```
1000 ' -----
1001 ' | REMBRANDTS TOOLKIT |
1002 ' -----
1003 '
1004 ' bij regel 1106 :
1005 '     J=0 : Cursortoetsenbesturing
1006 '     J=1 : Joystick nr. 1
1007 '     J=2 : Joystick nr. 2
1008 ' De screendumprountine werkt
1009 '     alleen bij MSX-printers en
1010 '     de Epson GX80 met MSX-kaart
1099 '
1100 ' -----
1101 ' | Initialisatie |
1102 ' -----
1103 COLOR 15,0,0
1104 MAXFILES=3
1105 S=-1:D=0:L=0:E=0:B=0:P=0:R=0
1106 J=0
1107 PA=BASE(12):CO=BASE(11)
1108 FOR I=0 TO 8:READ DX(I),DY(I):NEXT I
1109 DATA 0,0, 0,1, 1,1, 1,0, 1,-1, 0,-1, -1,-1,
-1,0, -1,1
1110 FOR I=1 TO 10:READ F$(I):NEXT I
1111 ON ERROR GOTO 2602
1112 DATA "F1 : Draw", "F2 : Line", "F3 : Block", "F
4 : Circle", "F5 : Paint", "F6 : Load", "F7 : Save",
"F8 : Dump", "F9 : Clear", "F10: Quit"
1113 CLS:PRINT:PRINT
1114 PRINT"Welke tekening wilt u editen ?":PRINT:
FILES "?????????.pic"
1115 PRINT:PRINT:PRINT" .pic"
1116 PRINTCHR$(30);:INPUT"filenaam ";FL$
1117 IF LEFT$(FL$,1)=". " THEN PRINT:PRINT"FOUT":G
OTO 1115
1118 SCREEN 2,0
1119 A$="":FOR I=1 TO 8:READ B$:A$=A$+CHR$(VAL("&
h"+B$)):NEXT I:SPRITE$(1)=A$
1120 DATA ff,c0,a0,90,88,84,82,81
1121 OPEN "grp:" FOR OUTPUT AS #1
1122 ON KEY GOSUB 1302,1402,1502,1602,1702,1802,1
902,2002,2102,2202
1123 ON STRIG GOSUB 2302,2302,2302,2302,2302
```

```

1124 FOR I=1 TO 10:KEY(I) ON:NEXT I
1125 FOR I=0 TO 4:STRIG(I) ON:NEXT I
1126 GOSUB 2102
1199 '
1200 ' -----
1201 ' | Invoerlus |
1202 ' -----
1203 X=X+DX(STICK(J)):Y=Y-DY(STICK(J))
1204 IF X<1 THEN X=1 ELSE IF X>150 THEN X=150
1205 IF Y<1 THEN Y=1 ELSE IF Y>190 THEN Y=190
1206 IF D THEN PSET(X,Y),15
1207 PUT SPRITE 1,(X,Y),14,1
1208 GOTO 1203
1299 '
1300 ' -----
1301 ' | F1 - Draw aan/uit |
1302 ' -----
1303 D=NOT(D)
1304 COLOR 6+D,0,0
1305 PRESET (160,10):PRINT#1,F$(1)
1306 GOSUB 2402:RETURN
1399 '
1400 ' -----
1401 ' | F2 - Line aan/uit |
1402 ' -----
1403 L=NOT(L)
1404 COLOR 6+L,0,0
1405 PRESET (160,20):PRINT#1,F$(2)
1406 LL=0
1407 GOSUB 2402:RETURN
1499 '
1500 ' -----
1501 ' | F3 - Block aan/uit |
1502 ' -----
1503 B=NOT(B)
1504 COLOR 6+B,0,0
1505 PRESET (160,30):PRINT#1,F$(3)
1506 BB=0
1507 GOSUB 2402:RETURN
1599 '
1600 ' -----
1601 ' | F4 - Circle aan/uit |
1602 ' -----
1603 E=NOT(E)
1604 COLOR 6+E,0,0
1605 PRESET (160,40):PRINT#1,F$(4)

```

```

1606 EE=0
1607 GOSUB 2402:RETURN
1699 '
1700 ' -----
1701 ' | F5 - Paint aan/uit |
1702 ' -----
1703 P=NOT(P)
1704 COLOR 6+P,0,0
1705 PRESET (160,50):PRINT#1,F$(5)
1706 GOSUB 2402:RETURN
1799 '
1800 ' -----
1801 ' | F6 - Load tekening |
1802 ' -----
1803 COLOR 5,0,0
1804 PRESET (160,60):PRINT#1,F$(6)
1805 BLOAD FL$,S
1806 COLOR 6,0,0
1807 PRESET (160,60):PRINT#1,F$(6)
1808 GOSUB 2106:GOSUB 2402:RETURN
1899 '
1900 ' -----
1901 ' | F7 - Save tekening |
1902 ' -----
1903 COLOR 5,0,0
1904 PRESET (160,70):PRINT#1,F$(7)
1905 BSAVE FL$,0,16383,S
1906 COLOR 6,0,0
1907 PRESET (160,70):PRINT#1,F$(7)
1908 GOSUB 2402:RETURN
1999 '
2000 ' -----
2001 ' | F8 - Screendump |
2002 ' -----
2003 COLOR 5,0,0
2004 PRESET (160,80):PRINT#1,F$(8)
2005 OPEN "LPT:" FOR OUTPUT AS #2
2006 GOSUB 2502
2007 COLOR 6,0,0
2008 PRESET (160,80):PRINT#1,F$(8)
2009 GOSUB 2402:RETURN
2099 '
2100 ' -----
2101 ' | F9 - Clear scherm |
2102 ' -----
2103 LINE (0,0)-(151,191),15,B

```

```

2104 LINE (1,1)-(150,190),8,BF
2105 T$=""
2106 S=-1:D=0:L=0:E=0:B=0:P=0:R=0
2107 COLOR 6,0,0
2108 FOR I=1 TO 10
2109 PRESET(160,I*10):PRINT#1,F$(I)
2110 NEXT I
2111 GOSUB 2402:RETURN
2199 '
2200 ' -----
2201 ' | F10 - Quit programma |
2202 ' -----
2203 IF INKEY$<>"" THEN 2203
2204 COLOR 15,0,0:END
2205 GOSUB 2402:RETURN
2299 '
2300 ' -----
2301 ' | Reageer op actie-toets |
2302 ' -----
2303 IF S THEN PSET(X,Y),15
2304 IF L AND LL THEN LINE(XX,YY)-(X,Y),15
2305 IF L AND NOT(LL) THEN XX=X:YY=Y:LL=L ELSE LL=N
OT(L)
2306 IF B AND BB THEN LINE(XX,YY)-(X,Y),15,B
2307 IF B AND NOT(BB) THEN XX=X:YY=Y:BB=B ELSE BB=N
OT(B)
2308 IF E AND EE THEN CIRCLE(XX,YY),SQR((X-XX)^2+
(Y-YY)^2),15
2309 IF E AND NOT(EE) THEN XX=X:YY=Y:EE=E ELSE EE=N
OT(E)
2310 IF P THEN PAINT(X,Y),15
2311 RETURN
2399 '
2400 ' -----
2401 ' | Diversen |
2402 ' -----
2403 S=NOT(D OR L OR E OR B OR P)
2404 RETURN
2499 '
2500 ' -----
2501 ' | Screendump routine |
2502 ' -----
2503 FOR I=144 TO 0 STEP -8:PRINT#2,CHR$(27);"T15
";CHR$(27);"S0192";
2504 FOR J=0 TO 5894 STEP 256
2505 FOR K=0 TO 7

```

```

2506 A=I+J+K
2507 BY=VPEEK(A+PA)
2508 IF BY=0 THEN IF (VPEEK(A+CO)AND15)=15 THEN B
Y=255
2509 PRINT#2, CHR$(BY);
2510 NEXT K:NEXT J
2511 PRINT#2, :NEXT I
2512 PRINT#2,
2513 CLOSE #2
2514 RETURN
2599 '
2600 ' -----
2601 ' | Error afhandeling |
2602 ' -----
2603 RESUME NEXT

```

#### 4.4 SCHAAKKLOK

Dit programma maakt uw schaakklok overbodig! Had u er nog geen dan is dit nu verholpen. Dit programma tovert een echte schaakklok op het scherm van de MSX.

Een schaakklok houdt bij hoeveel tijd iedere speler (bij het schaakspel natuurlijk) heeft zitten piekeren over zijn zetten. Voordat het spel begint wordt afgesproken hoeveel bedenktijd iedere speler krijgt. Gewoonlijk krijgt ieder twee en een half uur, maar bij een potje snelschaak kan dit ook twaalf minuten zijn.

Nadat we het RUN-commando hebben gegeven wordt naar de bedenktijd gevraagd. We geven deze op in uren en minuten. Vervolgens wordt naar de namen van de spelers gevraagd.

De schaakklok verschijnt in beeld, de linkerklok is van wit, de rechter van zwart. Wit begint en de linkerklok loopt dus. Als wit zijn zet gedaan heeft drukt hij even op de spatiebalk. Hierdoor wordt zijn klok stil gezet en gaat de klok van zwart lopen. Nu is zwart aan de beurt, voor hem geldt hetzelfde. Nadat hij gezet heeft drukt hij even op de spatiebalk om zijn klok stil te zetten en die van wit te starten.

Als de maximale speeltijd van een van de spelers is verstreken voor iemand mat is gezet heeft deze verloren, in schaaktermen: zijn vlag is gevallen.

*Programma:*

```

10 REM schaakklok
20 COLOR 15,4,4
30 KEY OFF
40 DEFSNG A-Z
50 A$="m10131s0"
60 B$="m600012"
70 PLAY A$,A$,A$
80 OPEN "grp:" FOR OUTPUT AS #1

```



```

90 ON STRIG GOSUB 810
100 ON INTERVAL=50 GOSUB 350
110 CLS:PRINTSPC(13)"SCHAAKKLOK"
120 LOCATE 10,10:PRINT"even geduld aub..
130 GOSUB 740'vul arrays
140 CLS:PRINTSPC(13)"SCHAAKKLOK"
150 PRINT"totale speeltijd per speler :
160 PRINT
170 INPUT"    uren";UM:U=UMOD12
180 INPUT"  minuten";MM
190 INPUT"naam speler wit    ";W$
200 INPUT"naam speler zwart ";Z$
210 SCREEN2
220 KP=1:GOSUB 440
230 KP=2:GOSUB 440:GOSUB 820
240 GOSUB 700'init
250 GOSUB 520' gr wijzer
260 GOSUB 570' kl wijzer
270 PSET(65,0),1:PRINT#1,"SPEELTIJD:"STR$(UM)": "R
IGHT$( "0"+MID$(STR$(MM),2),2)
280 PSET(50,13),4:PRINT#1,"WIT"
290 PSET(175,13),4:PRINT#1,"ZWART"
300 PSET(30,150),4:PRINT#1,W$
310 PSET(165,150),4:PRINT#1,Z$
320 INTERVAL ON
330 IF U>=UU AND M>=MM THEN GOSUB 1030'einde tijd
340 GOTO 320 ' hoofdlus
350 STRIG(0) STOP:S=(S+1)MOD60:GOSUB 620
360 IF S=M+1 THEN GOSUB 520
370 IF S=U*5+1 THEN GOSUB 570
380 IF S=0 THEN M=(M+1)MOD 60:GOSUB 520
390 IF M=0 AND S=0 THEN IB=U+1:U=(U+1)MOD12:GOSUB
570:PLAY B$,B$,B$
400 SZ=(SZ+1)MOD60
410 IF SZ=0 THEN MZ=(MZ+1)MOD60
420 GOSUB 1120
430 STRIG(0) ON:RETURN
440 'klok tekenen
450 IF KP=1 THEN X0=64 ELSE X0=192
460 Y0=80:R=8.8
470 CIRCLE(X0,Y0),6.3*R
480 FOR I=0 TO 55 STEP 5
490 A=6*R*AS(I):B=6*R*BC(I)
500 CIRCLE(X0+B,Y0+A),1
510 NEXT I
520 LINE (B0+X0,Y0+A0)-(X0,Y0),4

```

```

530 A=4*R*AS(M):B=4*R*BC(M)
540 LINE(X0,Y0)-(X0+B,Y0+A),15
550 A0=A:B0=B
560 RETURN
570 LINE(X0+D0,Y0+C0)-(X0,Y0),4
580 A=3*R*AS(U*5):B=3*R*BC(U*5)
590 LINE(X0,Y0)-(X0+B,Y0+A),15
600 C0=A:D0=B
610 RETURN
620 PLAY A$,A$,A$
630 PLAY"o6c","o7c","o8g"
640 LINE(X0+F0,Y0+E0)-(X0,Y0),4
650 A=5*R*AS(S):B=5*R*BC(S)
660 LINE(X0,Y0)-(X0+B,Y0+A),15
670 E0=A:F0=B
680 LINE(X0-2,Y0-1)-(X0+2,Y0+1),,BF
690 RETURN
700 A0=4*R*AS(M):B0=4*R*BC(M)
710 C0=3*R*AS(U*5):D0=3*R*BC(U*5)
720 E0=5.7*R*AS(D):F0=5.7*R*BC(D)
730 RETURN
740 DIM AS(60),BC(60)
750 FOR I=0 TO 60
760 D=(I-15)*.1047197#
770 AS(I)=SIN(D)
780 BC(I)=COS(D)
790 NEXT I
800 RETURN
810 'strig
820 INTERVAL STOP:STRIG(0) OFF
830 MZ=0:SZ=0
840 IF SS=0 THEN 860
850 IF (SS=1 AND KP=2) OR (SS=2 AND KP=1) THEN 99
0
860 IF KP=1 THEN 930
870 LINE(150,166)-(235,170),4,BF
880 LINE(20,166)-(100,170),3,BF
890 U2=U:M2=M:S2=S
900 U=U1:M=M1:S=S1
910 X0=64:GOSUB 1000
920 KP=1:GOTO 990
930 LINE(150,166)-(235,170),3,BF
940 LINE(20,166)-(100,170),4,BF
950 U1=U:M1=M:S1=S
960 U=U2:M=M2:S=S2
970 X0=192:GOSUB 1000

```

```

980 KP=2
990 STRIG(0) ON:RETURN
1000 GOSUB 700
1010 GOSUB 520:GOSUB 570:GOSUB 640
1020 RETURN
1030 INTERVAL OFF
1040 SS=KP
1050 IF SS=2 THEN 1070
1060 PSET(40,180),4:GOTO 1080
1070 PSET(160,180),4
1080 PRINT#1,"GESTOPT"
1090 PLAY B$+"O6C"
1100 IF PS=1 THEN 1100 ELSE PS=1
1110 GOSUB 820:RETURN
1120 LINE(100,129)-(160,138),4,BF
1130 PSET(101,130),4
1140 PRINT#1,STR$(MZ):"STR$(SZ)
1150 RETURN

```

#### 4.5 BUGGER

Bugger is de naam van een geweldig goede programmeerhulp. Dit programma kan u een lijst geven van alle gebruikte variabelen met bijbehorend regelnummer. Eveneens wordt informatie over de variabelen gegeven. Zo wordt een lijstvariabele (array) afgedrukt met "(" achter de naam. Het programma maakt onderscheid tussen integer, string en andere variabelen, precies zoals de MSX dat zou doen. Het is dus mogelijk snel te zien of een bepaalde variabele gebruikt is of nog gebruikt mag worden. Wordt een variabele slechts in één regel gebruikt, dan krijgt deze variabele kennelijk alleen een waarde en wordt verder niet gebruikt.

Dit programma kan echter nog meer, het kan u een lijst geven van alle sprongen die in het programma gedaan worden. Deze lijst bevat alle regelnummers waar naartoe gesprongen wordt en de regelnummers waar vandaan die sprong kwam. Dit is heel handig bij minder gestructureerde programma's. Of om snel inzicht te krijgen in de werking van een programma dat door een ander geschreven is.

#### WERKING

Nadat we het RUN-commando hebben gegeven wordt gevraagd naar de naam van de file die onderzocht moet worden. De file moet op een diskette staan en mag **niet** in ASCII-formaat gesaved zijn.

Tijdens het inlezen van de te onderzoeken file wordt steeds het regelnummer afgedrukt waar Bugger mee bezig is, evenals het hoeveelste byte is ingelezen. Wordt een variabele gevonden dan wordt de naam ervan ook afgedrukt.

Na het inlezen van de file wordt gevraagd hoe de gegevens uitgevoerd moeten worden:

(S)cherm, (P)rinter of (E)inde



```

90 IF N$="" THEN CLS:FILES:GOTO80
100 OPEN N$ AS #1 LEN=1
110 FIELD #1,1 AS BY$
120 GET 1:D$=BY$:IF ASC(D$)<>255 THEN PRINT:PRINT
"Dit is geen programma!":CLOSE:GOTO80
130 LF=LOF(1):PT=32769!
140 GOSUB 440:'inlezen
150 CLOSE
160 CLS:WIDTH40
170 LOCATE0,0:PRINT"***** UITVOER *****
*****"
180 LOCATE2,10:PRINT"(S)chem, (P)rinter of (E)in
de ";C$=INPUT$(1):S=-1
190 IF C$="p" THEN S=0
200 IF C$="e" THEN 380
210 LOCATE2,13:PRINT"(V)arlijst, (R)egellijst of
(B)eide ";C$=INPUT$(1):VT=0:LT=0
220 IF C$="v" THEN VT=-1
230 IF C$="r" THEN LT=-1
240 IF C$="b" THEN LT=-1:VT=-1
250 IF VT THEN GOSUB 1050
260 IF LT THEN GOSUB 1170
270 GOTO 160
280 LOCATE1,23:PRINT"stop=<ESC>  door=<SPATIE>";:
C$=INPUT$(1)
290 IF C$=" " THEN RETURN
300 IF C$=CHR$(27) THEN ERROR 101
310 'error
320 IF ERR=101 THEN RESUME 160
330 IF ERR=100 THEN 350
340 ON ERROR GOTO0:END
350 LOCATE2,16:PRINT"Einde van regel is niet gevo
nden
360 PRINT:PRINT"  in byte";LOC(1);"van de file"
370 PRINT:PRINT"  Sla een toets aan":C$=INPUT$(1)
:RUN
380 CLS:PRINT:PRINT"programma: ";N$
390 PRINT:PRINT"inlezen ? (y/n) ";
400 C$=INPUT$(1):IF C$<>"n" THEN CLS:LOAD N$
410 CLS:COLOR15,4,4:KEYON:NEW
420 END
430 REM inlezen programma
440 SCREEN0:WIDTH40:PRINT"***** INLEZEN F
ILE *****"
450 LOCATE 4,8:PRINT"programma: ";N$
460 LOCATE 4,10:PRINT"file is";LF;"bytes lang.

```

```

470 LOCATE 4,12:PRINT"regelnummer:"
480 LOCATE 4,14:PRINT"byte : "
490 GET1:P=ASC(BY$):GET1:P=256*ASC(BY$)+P
500 LOCATE11,14:PRINTLOC(1);STRING$(25,32)
510 IF P=0 THEN RETURN
520 GET1:R$=BY$:GET1:R$=BY$+R$
530 LOCATE 17,12:PRINTFNDR(R$);
540 LR=P-PT-5:PT=P:Q2$="*":Q1$="*":GOSUB580
550 GET1:IF BY$<>CHR$(0) THEN ERROR 100
560 GOTO 490
570 ' lezen regel
580 IF LR=0 THEN RETURN
590 GET1:LR=LR-1
600 P=PEEK(&H8006+ASC(BY$)-6*(ASC(BY$)>218))-48
610 IF P>9 THEN 650
620 LR=LR-P
630 IF P>0 THEN GET1,LOC(1)+P
640 GOTO580
650 ON P-16 GOTO 670,680,710,850,900
660 GOTO580
670 P=LR:GOTO620
680 IFLR=0 THEN RETURN
690 GET1:LR=LR-1:IF BY$<>CHR$(34) THEN 680
700 GOTO580
710 A$=BY$
720 IF LR=0 THEN 750
730 GET1:LR=LR-1:IF INSTR("ABCDEFGHIJKLMNQRSTU
WXYZ1234567890!#$%([]",BY$) THEN A$=A$+BY$:IF BY$
="(" THEN 750 ELSE 720
740 GET1,LOC(1)-1:LR=LR+1
750 IF INSTR(Q1$,"*"+A$+"*")<>0 THEN 580 ELSE Q1$
=Q1$+A$+"*"
760 S1=0:S2=MV:LOCATE11,14:PRINTSTRING$(25,32):LO
CATE11,14:PRINTLOC(1);" ";A$;
770 P=(S1+S2)\2:IF P=S2 THEN 820
780 IF VR$(P,0)<A$ THEN S2=P:GOTO 770
790 IF VR$(P,0)>A$ THEN S1=P+1:GOTO 770
800 IF RIGHT$(VR$(P,1),LEN(R$))=R$ THEN 580
810 VR$(P,1)=VR$(P,1)+R$:GOTO580
820 S2=MV:MV=MV+1
830 IF S2>P THEN VR$(S2,0)=VR$(S2-1,0):VR$(S2,1)=
VR$(S2-1,1):S2=S2-1:GOTO 830
840 VR$(P,0)=A$:VR$(P,1)=R$:GOTO580
850 VQ%=0
860 IF LR=0 THEN RETURN
870 GET1:LR=LR-1:IF BY$=CHR$(34) THEN VQ%=NOT VQ%

```

```

880 IF VQ% OR BY$<>":" THEN 860
890 GOTO580
900 IF LR=0 THEN RETURN
910 GET1:LR=LR-1:IF BY$=" " OR BY$="," THEN 900
920 IF BY$<>CHR$(14) THEN GET1,LOC(1)-1:LR=LR+1:G
OTO580
930 GET1:LR=LR-2:A$=BY$:GET1:A$=BY$+A$
940 IF INSTR(Q2$,"*"+A$+"*")<>0 THEN 900 ELSE Q2$
=Q2$+A$+"*"
950 S1=0:S2=MR
960 IFS2<S1 THEN 1020
970 P=(S1+S2)\2:IF P=S2 THEN 1020
980 IF RV$(P,0)<A$ THEN S2=P:GOTO 970
990 IF RV$(P,0)>A$ THEN S1=P+1:GOTO 970
1000 IF RIGHT$(RV$(P,1),LEN(R$))=R$ THEN 900
1010 RV$(P,1)=RV$(P,1)+R$:GOTO900
1020 S2=MR:MR=MR+1
1030 IF S2>P THEN RV$(S2,0)=RV$(S2-1,0):RV$(S2,1)
=RV$(S2-1,1):S2=S2-1:GOTO 1030
1040 RV$(P,0)=A$:RV$(P,1)=R$:GOTO900
1050 IF S THEN 1080
1060 LPRINT:LPRINT"VARIABELENLIJST VAN PROGRAMMA
";N$:LPRINT
1070 FOR A=MV-1 TO 0 STEP-1:LPRINTVR$(A,0);" ";:F
OR C=1 TO LEN(VR$(A,1)) STEP2:LPRINT FNZ$(MID$(VR
$(A,1),C,2));:NEXT C:LPRINT:NEXT A:LPRINT:LPRINT:
RETURN
1080 A=MV
1090 CLS:PRINT"***** VARIABELENLIJST *****
*****"
1100 A=A-1:IF A<0 THEN GOSUB280:RETURN
1110 B=LEN(VR$(A,0))+2:PRINT:PRINTVR$(A,0);:LOCAT
E B,CSRLIN
1120 FOR C=1 TO LEN(VR$(A,1)) STEP2
1130 PRINTFNZ$(MID$(VR$(A,1),C,2));:IF POS(1)>32
THEN PRINT:LOCATE B,CSRLIN
1140 NEXT C:PRINT
1150 IF CSRLIN>18 THEN GOSUB280:GOTO 1090
1160 GOTO1100
1170 IF S THEN 1200
1180 LPRINT:LPRINT"REGELSPRONGENLIJST VAN PROGRAM
MA ";N$:LPRINT
1190 FOR A=MR-1 TO 0 STEP-1:LPRINTFN Z$(RV$(A,0))
;"*";:FOR C=1 TO LEN(RV$(A,1)) STEP2:LPRINT FNZ$(
MID$(RV$(A,1),C,2));:NEXT C:LPRINT:NEXT A:LPRINT:
LPRINT:RETURN

```

```

1200 A=MR
1210 CLS:PRINT"***** REGELSPRONGENLIJST *****
*****
1220 A=A-1:IF A<0 THEN GOSUB280:RETURN
1230 B=LEN(STR$(FNDR(RV$(A,0))))+1:PRINT:PRINTFNZ
$(RV$(A,0));"< ";:LOCATE B,CSRLIN
1240 FOR C=1 TO LEN(RV$(A,1)) STEP2
1250 PRINTFNZ$(MID$(RV$(A,1),C,2));:IF POS(1)>32
THEN PRINT:LOCATE B,CSRLIN
1260 NEXT C:PRINT
1270 IF CSRLIN>19 THEN GOSUB280:GOTO 1210
1280 GOTO1220

```

#### 4.6 CHECKSUMGENERATOR

Een checksumgenerator is een programma dat per programmaregel een getal genereert dat direct gekoppeld is aan de inhoud van die regel. Is de inhoud van de regel anders dan zal ook de checksum anders zijn. Zo is het mogelijk aan de hand van de uitkomsten van de checksumgenerator na te gaan of het programma goed is ingetoetst en zoniet, waar dan de fouten zitten.

Het is dan natuurlijk wel noodzakelijk te weten wat de checksums (controlegetallen) van de originele versie waren.

In dit boek hebben wij de tekstverwerker en de database voorzien van de checksums per regel, zoals die door het onderstaande checksumprogramma gegenereerd zijn. Het is dus eenvoudig te controleren of alles goed is ingetoetst.

Deze checksumgenerator neemt in de checksum de regelnummers mee. Dus een ander regelnummer levert ook een andere checksum op.

Verwisselde letters worden ook ontdekt. Dus da levert een andere checksum dan ad.

#### WERKWIJZE

De file waarvan de checksums bepaald moeten worden, moet eerst als ASCII gesaved worden. Dit doen we als volgt:

```
SAVE FILENAAM,A
```

Hierna moet het programma „checksum“ geladen worden. Na het geven van het RUN-commando komt het programma met de vraag: „filenaam“. Nu moet de naam van de zojuist gemaakte ASCII-file ingetypt worden.

#### *Opmerking:*

Standaard wordt diskdrive A: genomen, indien van DiskBASIC gebruik gemaakt wordt. Wordt van CassetteBASIC gebruik gemaakt, wordt natuurlijk de cassette als standaard randapparaat genomen.



Als we nu vanuit DiskBASIC een programma van cassette willen gebruiken dan moeten we voor de filenaam CAS: intypen, dus:

CAS:filenaam.asc

Vervolgens verschijnt de vraag „Printer?“ op het scherm. Hierop kan met j of n geantwoord worden, beide gevolgd door RETURN. Als met j(a) geantwoord wordt verschijnt alle uitvoer zowel op het scherm als op de aangesloten printer. Wordt met n(ee) geantwoord dan verschijnt alles alleen op het scherm.

Het programma berekent twee checksums:

- een blokchecksum en
- een regelchecksum

Door deze beide checksums is het eenvoudig een programma op fouten te controleren en te achterhalen waar de fout is opgetreden.

Bij een aantal programma's in dit boek hebben we het resultaat van het checksumprogramma afgedrukt. Dus per programmaregel is een checksum gegeven. Let wel, ook het regelnummer wordt meegenomen in deze checksum. Het is dus zaak exact dezelfde nummering aan te houden.

Onder de listing zien we de regelnummers met hun checksum. Als we nu het ingetypte programma op bovenstaande manier door het checksumprogramma halen, krijgen we als we geen typfouten gemaakt hebben, dezelfde checksums als afgedrukt in het boek.

Klopt één van de getallen niet met het getal in het boek, dan hebben we een typfout in die regel gemaakt.

#### **Opmerking:**

Natuurlijk is het niet mogelijk aan de hand van checksums altijd alle typfouten uit een programma te halen. Het blijft mogelijk dat typfouten optreden die niet ontdekt worden, omdat toevallig toch dezelfde checksum verkregen wordt. Deze kans is echter uiterst klein omdat regel- en blokchecksum onafhankelijk van elkaar bepaald worden.

#### **Programma:**

```
10 '*****
20 '** Checksumgenerator V 1.0 **
30 '** Voor MSX. Door J.Ribberink**
40 '*****
50 CLEAR 4000 : MAXFILES=2
60 SW=39 : PW=60 : GR=16 : KK=0 : KP=0
70 A$="a" : WIDTH(SW+1)
80 '*****
90 '** Gegevensinvoer **
100 '*****
110 CLS : LOCATE 7,5
120 PRINT"CHECKSUM-GENERATOR"
```

```

130 LOCATE 7,10 : LINEINPUT"Filenaam ? ";F$
140 LOCATE 7,12 : INPUT"Printer";A$
150 IF A$="j" THEN PR=-1 ELSE PR=0
160 CLS : IF PR THEN LPRINT"checksum van ";F$
170 '*****
180 '** Hoofdprogramma **
190 '*****
200 OPEN F$ FOR INPUT AS #2
210 SUM=0 : GTL=-1 : B$=""
220 IF NOT EOF(2) THEN A=ASC(INPUT$(1,#2))ELSE 34
0
230 IF GTL THEN GTL=(A>47 AND A<58) : IF GTL THEN
B$=B$+CHR$(A)
240 IF A=32 THEN 220
250 IF A=13 THEN 300
260 SUM=SUM*2+A : SSUM=SSUM/2+A
270 IF SSUM-INT(SSUM) THEN SSUM=INT(SSUM)+32768!
280 IF SUM>65535! THEN SUM=SUM-65535! : GOTO 280
290 GOTO 220
300 B$=RIGHT$(SPACE$(4)+B$,5)
310 K=K+1 : M=1 : GOSUB 400
320 IF K>=GR THEN K=0 : M=2 : GOSUB 400 : SSUM=0
330 IF NOT EOF(2) THEN A=ASC(INPUT$(1,#2)): GOTO
210
340 CLOSE #2
350 IF K<>0 THEN M=2 : GOSUB 400
360 END
370 '*****
380 '** Druk af op scherm/printer **
390 '*****
400 PA=SW-11 : OPEN"CRT:" FOR OUTPUT AS #1
410 ON M GOSUB 500,570 : CLOSE #1
420 IF NOT PR THEN 460 ELSE KS=KK : KK=KP
430 PA=PW-11 : OPEN"lpt:" FOR OUTPUT AS #1
440 ON M GOSUB 500,570 : CLOSE #1
450 KP=KK : KK=KS
460 RETURN
470 '*****
480 '** Maak afdruk **
490 '*****
500 IF KK>PA THEN KK=0 : PRINT #1,""
510 KK=KK+11 : PRINT#1,LEFT$(B$,5);": ";
520 PRINT#1,RIGHT$(STRING$(3,"0")+HEX$(SUM),4);"
";
530 RETURN

```

```
540 '*****
550 '** Druk blokchecksum af      **
560 '*****
570 IF SSUM>65535! THEN SSUM=SSUM-65536! : GOTO 5
70
580 IF KK>PA-9 THEN PRINT#1,"" : KK=0
590 PRINT #1,SPACE$(PA-10-KK);"blok-checksum : ";
600 PRINT #1," ";RIGHT$(STRING$(3,"0")+HEX$(SSUM)
,4)
610 KK=0 : RETURN
```

# 5 SIMULATIE

## 5.1 INLEIDING

De kracht van het simuleren van processen op computersystemen wordt door steeds meer wetenschappers ontdekt. Immers, door het wijzigen van het model of het variëren van de parameters kan men onderzoeken hoe een proces zich onder verschillende omstandigheden gedraagt, zonder dat daarvoor ingewikkelde en kostbare proefopstellingen nodig zijn. In dit hoofdstuk presenteren we twee simulatieprogramma's. Het eerste, een programma voor analoge simulatie, is gebaseerd op het THTSIM-programma, een algemeen simulatieprogramma dat op de Technische Hogeschool Twente werd ontwikkeld. Uitgebreide versies van het THTSIM programma zijn onder meer onder de naam TUTSIM in de handel gebracht.

Het tweede programma is een programma voor digitale simulatie, ontwikkeld door Chris Baten, een student aan de TH Twente. Met het programma kunnen zogenaamde combinatorische digitale schakelingen worden gesimuleerd.

Karakteristiek voor beide programma's is dat een model wordt opgebouwd met behulp van "blokjes" die specifieke functies voorstellen. Ieder blokje heeft steeds één uitgang, terwijl het aantal ingangen kan variëren.

Om tot simulatie van een bepaald systeem te komen, hanteren we de volgende werkwijze:

- bepaal een blokschema voor het te simuleren systeem
- voer de structuur van het model in en geef de parameters aan
- voer de gewenste "simulatie-runs" uit.

We zullen voor beide programma's de gehele procedure aan de hand van een concreet voorbeeld toelichten.

## 5.2 ANALOGE SIMULATIE

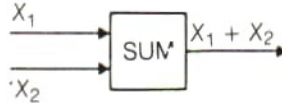
Het analoge simulatieprogramma leent zich uitstekend voor het simuleren van zeer uiteenlopende dynamische processen. Niet alleen mechanische systemen kunnen worden gesimuleerd, maar ook chemische, biologische en vele andere.

### 5.2.1 DE BOUWSTENEN

Om een zogenaamd simulatienetwerk van een bepaald systeem op te zetten, mogen we de volgende functionele eenheden gebruiken:

## OPTELLER (SUMMATOR)

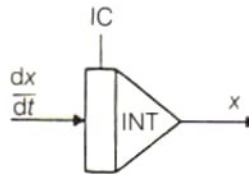
De „opteller” heeft twee ingangssignalen en één uitgangssignaal; het uitgangssignaal is per definitie de som van beide ingangssignalen.



We tekenen de opteller als een blokje met 2 ingangen en 1 uitgang. Binnen het blokje plaatsen we de naam SUM om de functie „opteller” aan te geven.

## INTEGRATOR

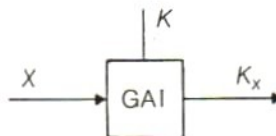
De integrator draagt als kenmerk dat het uitgangssignaal de integraal is van het ingangssignaal plus de zogenaamde beginwaarde. We zullen straks zien dat integratoren een grote rol spelen bij het simuleren van processen die door differentiaalvergelijkingen worden beschreven.



Merk op dat hier als ingangssignaal de afgeleide van het uitgangssignaal is aangegeven. De reden hangt samen met het feit dat we doorgaans met differentiaalvergelijkingen te maken hebben waar deze afgeleiden als zodanig in genoteerd zijn. Aan de hand van het straks te bespreken voorbeeld zal dit nog wel duidelijker worden.

## VERSTERKINGSFACTOR (GAIN)

Deze functionele eenheid draagt als kenmerk dat het uitgangssignaal steeds overeenkomt met het ingangssignaal vermenigvuldigd met een bepaalde constante. Deze constante duiden we ook wel eens aan met „gain factor”, „versterkingsfactor” of „proportionele factor”

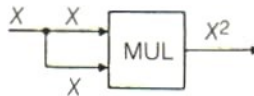


## VERMENIGVULDIGER (MULTIPLIER)

We hebben hier wederom te maken met een blok met 2 ingangssignalen; het uitgangssignaal is nu het product van beide ingangssignalen.



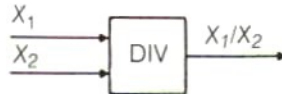
Merk op dat we met dit blokje ook eenvoudig het kwadraat van een zekere waarde kunnen bepalen. Het volgende schema toont de oplossing.



We zullen straks zien hoe we deze situatie door een speciale vorm van noteren kunnen aangeven.

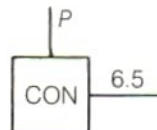
## DELER (DIVIDER)

Deze eenheid bepaalt het quotiënt van twee ingangssignalen:



## CONSTANTE

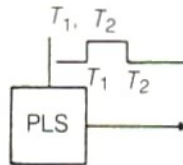
Een eenheid die veel in simulatieschema's wordt gebruikt en wel om bepaalde vaste waarden (constanten) aan te geven. We hebben hier te maken met een blok dat geen ingang heeft.



De parameter  $p$  bepaalt de constante waarde van het uitgangssignaal.

## PULS

De volgende bouwsteen dient om het gedrag van een systeem als resultaat van bepaalde variaties in tijd te kunnen bestuderen. Dit blok genereert een puls met amplitude 1 en wel tussen de aangegeven momenten  $T_1$  en  $T_2$ . De amplitude van de puls kan men uiteraard eenvoudig met een GAI-functie instellen.



Door een aantal van deze PLS-blokken met daaraan gekoppelde GAI-blokken parallel te plaatsen waarbij de som uiteindelijk door een aantal SUM-blokken wordt verkregen kunnen we trouwens allerlei variaties beschrijven.

## TIJD (TIME)

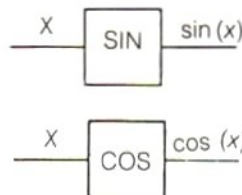
De belangrijkste grootheid bij allerlei simulaties is de tijdfunctie. Deze wordt steeds gesimuleerd met een functie waarvan de uitgang lineair toeneemt.



De mate waarin de tijd toeneemt en de momenten waarop we de uitgangswaarden van de verschillende blokken kunnen bestuderen, worden aan de hand van nog te bespreken commando's ingesteld.

## SIN, COS EN EXP

Deze blokken zijn beschikbaar om de bekende functies sinus, cosinus en exponent (e-macht) van een ingangssignaal te bepalen.





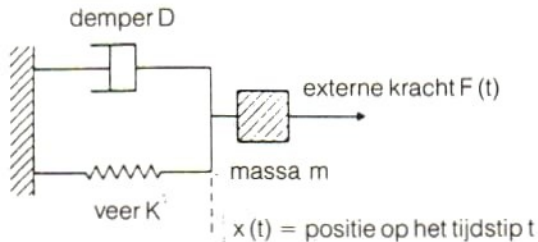
### 5.2.2 EEN UITGEWERKT VOORBEELD

Om te illustreren hoe met deze elementaire blokken het gedrag van bepaalde systemen kan worden gesimuleerd beschouwen we een massa-veer-demper systeem. De demonstratie bestaat uit de volgende drie fasen:

1. Tonen hoe men op grond van de gegeven differentiaalvergelijking tot een simulatieschema komt, waarin van de bovengenoemde elementaire functies gebruik wordt gemaakt.
2. Tonen hoe een dergelijk schema op de computer kan worden ingevoerd waarbij dan gebruik wordt gemaakt van het simulatie-programma.
3. Het daadwerkelijk werken met het simulatie-programma, kortom het uitvoeren van concrete simulaties.

Ad 1. Opstellen van het simulatieschema.

We gaan als voorbeeld van het volgende mechanische massa-veer-demper systeem uit:



Als  $F(t)$  de kracht is die op deze massa werkt dan zal het dynamische evenwicht worden bepaald door de volgende reactiekrachten.

massatraagheidskracht:  $M \frac{d^2 x}{d t^2}$

demping:  $D \frac{dx}{dt}$

veerkracht:  $K x$



De volgende vergelijking geldt:

$$M \frac{d^2 x}{dt^2} + D \frac{dx}{dt} + K x = F(t)$$

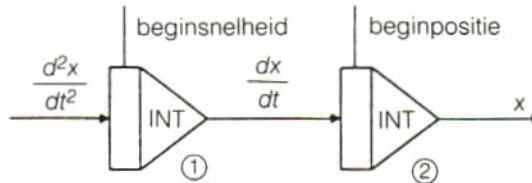
We merken hierbij op dat deze vergelijking met een differentiaalvergelijking overeenkomt en een dergelijke vergelijking kan dan, mits  $F(t)$  niet al te bizarre vormen aanneemt, analytisch worden opgelost. Bij het werken met het simulatieprogramma hoeven we echter niet naar zo'n analytische oplossing te zoeken; we zullen de vergelijking als het ware met onze elementaire blokjes nabouwen. Het aardige is dan dat we om tot een bepaalde oplossing te komen (en wel de oplossing die behoort bij een bepaalde  $F(t)$ , bijvoorbeeld een puls) geen verdere wiskundige kennis nodig hebben!

Om tot het juiste simulatieschema te komen, herschrijven we de gegeven vergelijking eerst zodanig dat de hoogste afgeleide links van het gelijkteken komt te staan. Voor de gegeven vergelijking leidt dit tot:

$$\frac{d^2 x}{dt^2} = \frac{F(t) - D \frac{dx}{dt} - K x}{M}$$

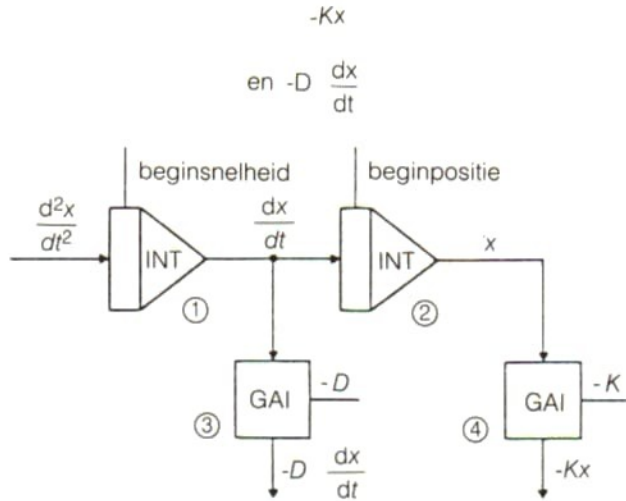
Merk op dat hier een gelijkheid staat en wel tussen de tweede afgeleide van  $x$  en de som van een aantal grootheden waarin o.a. de afgeleide van  $x$  en  $x$  zelf voorkomt.

Bij het simulatieschema zullen we allereerst van de volgende structuur uitgaan:

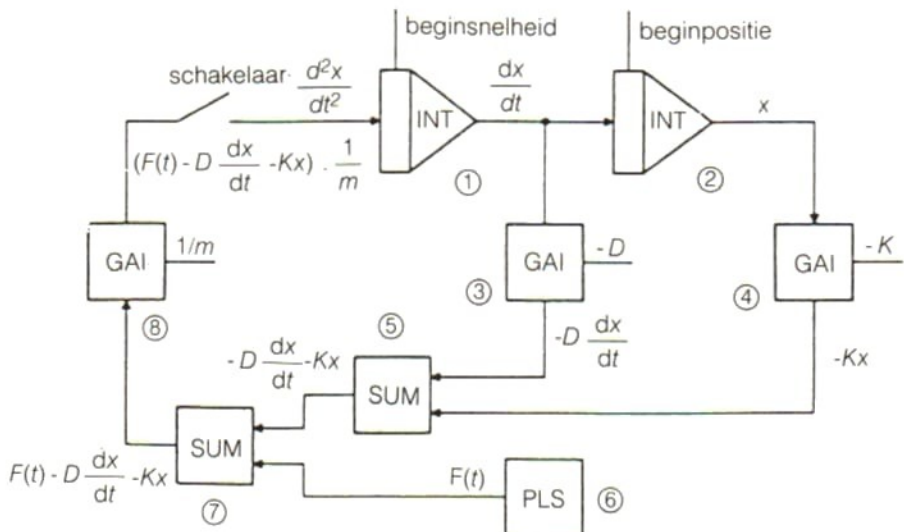


De „truc" is nu dat we uitgaande van de uitgangen van beide integratoren het rechterdeel van de zojuist gegeven vergelijking nabouwen. Tenslotte zullen we het zo verkregen netwerk verbinden met de ingang van de eerste integrator; de differentiaalvergelijking stelt immers dat de tweede afgeleide gelijk is aan het gedeelte rechts van het gelijkteken. Een en ander klinkt op dit moment wellicht nog wat moeilijk....we zullen echter zien hoe eenvoudig alles in z'n werk gaat.

Aan de uitgangen van beide integratoren verbinden we twee GAI-blokken om zo de volgende grootheden te bepalen:



Met behulp van een SUM-functie kunnen we deze beide grootheden optellen. Dit resultaat moet wederom met behulp van een SUM-functie worden opgeteld en wel bij de functie  $F(t)$ . Als functie  $F(t)$  zullen we als voorbeeld een puls-functie beschouwen. Het uiteindelijk resultaat moet gedeeld worden door  $M$  of, dat is het zelfde, vermenigvuldigd met  $1/M$ . Hiervoor gebruiken we wederom de GAI-functie. Zo verkrijgen we tenslotte het volgende schema:



Als de schakelaar gesloten wordt en  $F(t)$  geeft inderdaad een pulsvormig signaal af dan zal de uitgang van de tweede integrator de positie van ons massa-veer-demper systeem weergeven.

Merk nogmaals op dat de differentiaalvergelijking niet wordt opgelost maar als het ware met blokjes wordt nagebouwd.

Merk tenslotte op dat we ieder blok nu genummerd hebben; het nummer van zo'n blok stelt als het ware het identiteitsnummer voor. We zullen zo dadelijk zien waarvoor dit nummer gebruikt wordt.

## Ad 2. Notatie en invoer van het simulatieschema

Om de wijze van notatie goed te kunnen voorzien bedenken we dat ieder blok maar één uitgang heeft en tevens dat ieder blok ook één uniek nummer heeft gekregen.

Zo wordt ieder blok als onderdeel van het totale schema voorgesteld door de volgende notatie:

bloknummer; type; ingangen

Blok 5 is bijvoorbeeld een SUM en de beide ingangen zijn de uitgangen van de blokken 3 en 4. Zo wordt blok 5 als volgt aangegeven:

5;SUM;3;4

Ziezo, gewapend met deze kennis kunnen we nagaan hoe de hele structuur moet worden ingevoerd. We gaan er van uit dat ons simulatieprogramma in het geheugen van de computer is geplaatst.

Na het RUN-commando verschijnt de vraag:

Moet data ingelezen worden?

Omdat we nog niet eerder een structuur hebben ingevoerd en dus ook nog geen structuur op diskette hebben gesaved, drukken we alleen op ENTER.

Dan verschijnt de term STRUKTUUR ten teken dat nu de structuur, d.w.z. het simulatieschema ingevoerd moet worden. Vervolgens typen we alle blokken in, de volgorde maakt daarbij niet uit. Een bloknummer dat fout wordt ingevoerd kan daarna gewoon opnieuw ingevoerd worden. Als het laatste blok is ingevoerd toetsen we nogmaals op de ENTER-toets ten teken dat het volledige schema is ingevoerd.

Voor het voorbeeld:

### STRUKTUUR

```
3;GAI;1
1;INT;8
7;SUM;5;6
2;INT;1
4;GAI;2
8;GAI;7
5;SUM;3;4
6;PLS
S: ?
```

Hierna toont de computer de term PARAMETERS ten teken dat nu de parameters moeten worden ingevoerd. Hierbij wordt steeds de volgende notatie aangehouden:

bloknummer; parameters

Voor het voorbeeld stellen we als beginvoorwaarde dat de beginuitwijking en de beginsnelheid 0 zijn; dit houdt in dat de parameters van beide integratoren gelijk aan 0 zijn. Voor de overige parameters worden de volgende waarden gekozen:  $K = 1$ ,  $D = 0.1$  en  $M = 0.2$  ergo  $1/M = 5$ . Als parameters voor de puls nemen we 0 (beginpunt) en 1 s (eindpunt). Zo worden uiteindelijk de volgende parameters ingetoetst:

### PARAMETERS

```
1;0
2;0
3;-0.1
4;-1
6;0.0;1.0
8;5
P: ?
```

Na het invoeren van de laatste waarde wordt wederom op de ENTER-toets gedrukt ten teken dat alle parameters zijn ingevoerd.

Op dit moment is de volledige structuur ingevoerd. We dienen de computer nu nog op te geven welke grootheid we geplotted willen hebben en ook binnen welk interval. De computer vraagt daartoe eerst PLOTBLOCKS en we voeren in 2 en 6, we willen immers de uitwijking (blok 2) en de puls (blok 6) als functie van de tijd zien.

### PLOTBLOCKS

```
2;6
```

Vervolgens vraagt de computer om de PLOTRANGE m.a.w. om een minimale en maximale waarde waarbinnen de gevraagde grootheden zullen worden geplotted. Deze minimale en maximale waarde worden gewoonlijk proefondervindelijk bepaald. Liggen ze te ver uiteen dan zien we als resultaat een vrijwel rechte lijn. Zijn ze te klein dan zal de plotter zoals dat heet „in de verzadiging lopen (vastlopen)”. We voeren min of meer arbitrair als minimale waarde 1.2 en als maximale waarde 1.7 in.

### PLOTRANGE

```
-1.2;1.7
```

Als laatste gegeven vraagt de computer om de zogenaamde timing-gegevens. Deze gegevens omvatten een tweetal waarden: de eerste waarde geeft de stapjes **delta-t** aan waarvoor het systeem wordt doorgerekend en de tweede waarde bevat de eindtijd voor de betreffende simulatie. Als **delta-t** kiezen we 0.045 en als eindtijd 10 s.

## TIMING

T: ? Ø.Ø4;1Ø;2

Ad 3.De feitelijke simulatie

Op dit moment kan de feitelijke simulatie beginnen. We beschikken daartoe over een aantal commando's n.l.

CS verander structuur  
CP verander parameters  
CB verander plotblocks  
CR verander plotrange  
CT verander timing  
SA opstarten  
SD Graf. uitvoer (scherm)  
ST tabel uitvoer (scherm)  
HD Graf. uitvoer (printer)  
HT tabel uitvoer (printer)  
PR print data  
WR schrijf data naar diskette  
RE lees data van diskette  
H Help  
E einde

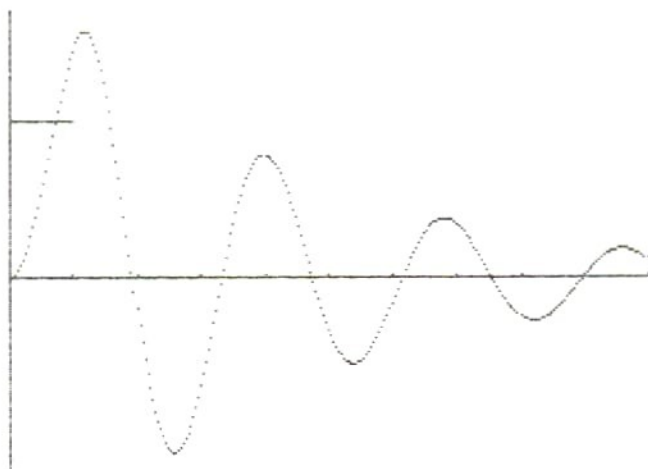
We geven ter controle nu eerst een PR-commando. Dat levert het volgende resultaat:

Structuur			Parameters	
1	INT	8		Ø
2	INT	1		Ø
3	GAI	1		-.1
4	GAI	2		-1
5	SUM	3	4	
6	PLS			Ø 1
7	SUM	5	6	
8	GAI	7		5

Output blocks : 2 6  
Max. Amplitude : 1.7  
Min. amplitude : -1.2  
Total time : 1Ø sec  
Stapgrootte SD : .Ø4 sec  
Stapgrootte ST : 2 sec

Commando: ?

Alles blijkt te kloppen. We kunnen nu gaan „tekenen“. We geven het commando SD en krijgen de volgende uitvoer:



Met dit programma kunnen we een veelheid van problemen en processen simuleren. Dit voorbeeld is één van de vele mogelijkheden.

*Programma:*

```
10 ' *** MSXSIM ***
20 CLEAR 1000:WIDTH 39
30 KEY OFF:DEFSNG A-Z
40 ON ERROR GOTO 4050
50 MA=350:DIM A(MA,4),P(4),B(MA),S(MA),FX(10),FFX
(10),RC(10),RL(3)
60 H$=STRING$(4,194)+CHR$(197):HL$=CHR$(195)
70 V$=CHR$(13)+CHR$(195):VL$=CHR$(195)
80 FOR I=1 TO 16:HL$=HL$+H$:NEXT
90 FOR I=1 TO 23:VL$=VL$+V$:NEXT
100 HL$=LEFT$(HL$,79)
110 C$="TIMCONPLSINTGAIRELFNCADLSINCOSEXPSSUBMU
LDIV"
120 D$="222333433334444":S$=SPACE$(20)
130 CLS:INPUT"Moet de data ingelezen worden (j/n)
";A$:GOSUB 1840
140 CLS:IF A$="J" THEN GOSUB 2260:GOTO 240
150 E=0:GOSUB 650 ' structuur
160 GOSUB 1920' sorteer
170 GOSUB 980 ' parameters
180 GOSUB 1200' plotblocks
```

```

190 GOSUB 1330' plotrange
200 GOSUB 1470' timing
210 GOSUB 1630' help
220 '-----
230 ' Commando interpreter
240 PRINT:INPUT"Commando: ";A$:A$=LEFT$(A$,2):GOSU
B 1840
250 IF A$="CS" THEN GOSUB 660:GOSUB 1920:GOTO 240
260 IF A$="CP" THEN GOSUB 980:GOTO 240
270 IF A$="CB" THEN GOSUB 1200:GOTO 240
280 IF A$="CR" THEN GOSUB 1330:GOTO 240
290 IF A$="CT" THEN GOSUB 1470:GOTO 240
300 IF A$="ST"OR A$="SD"OR A$="HT"OR A$="HD" THEN
GOSUB 2920:GOTO 240
310 IF A$="PR" THEN GOSUB 2020:GOTO 240
320 IF A$="SA" THEN GOTO 130
330 IF A$="WR" THEN GOSUB 2450:GOTO 240
340 IF A$="RE" THEN GOSUB 2260:GOTO 240
350 A$=LEFT$(A$,1):IF A$="E" THEN END
360 IF A$="H" THEN GOSUB 1630:GOTO 240
370 PRINT"Typ (H)elp aub..":GOTO 240
380 '-----
390 ' toets aangerakt?
400 IF INKEY$="" THEN RETURN
410 BEEP
420 IF INKEY$="" THEN 420
430 RETURN
440 '-----
450 BEPALEN VD PLAATS VAN ;
460 J=INSTR(I,A$,";")
470 IF J=0 THEN J=LEN(A$)+1
480 RETURN
490 '-----
500 ' ingangssignalen bekend?
510 J2=1
520 IF L=1 OR J2=J THEN RETURN
530 IF A(B(J),K)=A(B(J2),1) THEN L=1
540 J2=J2+1:GOTO 520
550 '-----
560 ' Zet scherm op
570 CLS:PRINT VL$;:IF Z%>=0 AND Z%<24 THEN LOCATE
0,Z%:PRINHL$;
580 LOCATE 0,24:PRINT"Schaal=Hor.: "D1"sec VER. "FA
C";
590 X=0:RETURN
600 '-----

```

```

610 ' Berichtje
620 BEEP:LOCATE 15,24:INPUT"Typ return aub";A$
630 RETURN
640 '-----
650 ' Invoer structuur
660 PRINT
670 PRINT"STRUKTUUR"
680 PRINT
690 NS=-1
700 A$="":INPUT"S: ";A$:GOSUB 1840
710 IF A$="" OR E>=MA THEN RETURN
720 E=E+1:I=1:GOSUB 460
730 A(E,1)=VAL(LEFT$(A$,J-I))
740 M=E:GOSUB 920
750 I=J+1:GOSUB 460
760 IF J-I<>3 THEN 910
770 G$=MID$(A$, I, 3):K=-2
780 K=K+3:IF K<46 AND G$<>MID$(C$,K,3)THEN 780
790 IF K>45 THEN 910
800 K=(K+2)/3:L=VAL(MID$(D$,K,1))
810 A(M,2)=K
820 IF L=2 THEN 880
830 FOR N=3 TO L
840 I=J+1:IF I>LEN(A$) THEN 910
850 GOSUB 460
860 A(M,N)=VAL(MID$(A$, I, J-I))
870 NEXT N
880 IF K=6 OR K=7 THEN 3890:' fu/rel
890 PRINTCHR$(30);LEFT$(A$,J-1);S$
900 GOTO 700
910 BEEP:PRINTCHR$(30);:GOTO 700
920 DU=A(E,1):IF E=1 THEN RETURN
930 FOR N=1 TO E-1
940 IF A(N,1)=DU THEN M=N:E=E-1:GOTO 960
950 NEXT N
960 RETURN
970 '-----
980 ' Invoer parameters
990 PRINT
1000 PRINT"PARAMETERS"
1010 PRINT
1020 A$="":INPUT"P: ";A$
1030 IF A$="" THEN RETURN
1040 I=1:GOSUB 460
1050 K=VAL(LEFT$(A$,J-1)):N=0
1060 N=N+1:IF N<E AND A(N,1)<>K THEN 1060

```



```

1070 IF A(N,1)<>K THEN 1170
1080 K=A(N,2):IF K=3 THEN 1130
1090 I=J+1:IF I>LEN(A$) OR K=1 OR K>6 THEN 1170
1100 GOSUB 460:A(N,4)=VAL(MID$(A$,I,J-I))
1110 PRINTCHR$(30);LEFT$(A$, (J-1));S$
1120 GOTO 1020
1130 I=J+1:IF I>LEN(A$) THEN 910
1140 GOSUB 460
1150 A(N,3)=VAL(MID$(A$,I,J-I))
1160 GOTO 1090
1170 BEEP:PRINTCHR$(30);:GOTO 1020
1180 RETURN
1190 '-----
1200 ' Invoer plotblocks
1210 PRINT
1220 PRINT"PLOTBLOCKS"
1230 PRINT
1240 A$="":INPUT"B: ";A$:I=1:Q=0
1250 IF Q=4 OR I>LEN(A$) THEN 1290
1260 GOSUB 460:Q=Q+1
1270 P(Q)=VAL(MID$(A$,I,J-I))
1280 I=J+1:GOTO 1250
1290 IF Q=0 THEN BEEP:PRINTCHR$(30);:GOTO 1240
1300 PRINTCHR$(30);LEFT$(A$,J);S$
1310 RETURN
1320 '-----
1330 ' Invoer plotrange
1340 PRINT
1350 PRINT"PLOTRANGE"
1360 PRINT
1370 A$="":INPUT"R: ";A$
1380 I=1:IF A$="" THEN 1440
1390 GOSUB 460:R=VAL(LEFT$(A$,J-1)):I=J+1
1400 IF I>LEN(A$) THEN 1440
1410 GOSUB 460:U=VAL(MID$(A$,I,J-I)):I=J+1
1420 PRINTCHR$(30);LEFT$(A$,J-1);S$
1430 RETURN
1440 BEEP:PRINTCHR$(30);:GOTO 1370
1450 RETURN
1460 '-----
1470 ' Invoer timing
1480 PRINT
1490 PRINT"TIMING"
1500 PRINT
1510 A$="":INPUT"T: ";A$
1520 I=1:IF A$="" THEN 1600

```

```

1530 GOSUB 460:D1=VAL(LEFT$(A$,J-1)):I=J+1
1540 IF I>LEN(A$) THEN 1600
1550 GOSUB 460:T1=VAL(MID$(A$,I,J-I)):I=J+1
1560 IF I>LEN(A$) THEN T2=D1:GOTO 1580
1570 GOSUB 460:T2=VAL(MID$(A$,I,J-I))
1580 PRINTCHR$(30);LEFT$(A$,J-1);S$
1590 RETURN
1600 BEEP:PRINTCHR$(30);:GOTO 1510
1610 RETURN
1620 '-----
1630 ' Invoer help
1640 CLS:PRINT"MSXSIM - HELP"
1650 PRINT
1660 PRINT"Commandolijst : "
1670 PRINT
1680 PRINT"CP verander parameters."
1690 PRINT"CB verander plotblocks."
1700 PRINT"CR verander plorange."
1710 PRINT"CT verander timing."
1720 PRINT"SA opstarten."
1730 PRINT"SD Graf. uitvoer.(scherm)"
1740 PRINT"ST tabel uitvoer.(scherm)"
1750 PRINT"HD Graf. uitvoer.(printer)"
1760 PRINT"HT tabel uitvoer.(printer)"
1770 PRINT"PR print data."
1780 PRINT"WR schrijf data naar diskette."
1790 PRINT"RE lees data van diskette."
1800 PRINT"H help"
1810 PRINT"E einde."
1820 PRINT
1830 RETURN
1840 '-----
1850 ' Omzetting in grote letters
1860 IF LEN(A$)=0 THEN 1910
1870 FOR I%=1 TO LEN(A$)
1880 HP%=ASC(MID$(A$,I%,1))
1890 MID$(A$,I%,1)=CHR$(HP%+((HP%>96) AND (HP%<12
2))*32)
1900 NEXT
1910 RETURN
1920 ' sorteer structuur
1930 IF E=1 THEN RETURN
1940 FOR I=1 TO E-1
1950 DU=A(I,1):DI=I
1960 FOR N=I+1 TO E
1970 IF A(N,1)<DU THEN DU=A(N,1):DI=N

```

```

1980 NEXT N
1990 FOR N=1 TO 4:SWAP A(I,N),A(DI,N):NEXT N
2000 NEXT I:RETURN
2010 '-----
2020 ' Print data op scherm
2030 CLS:PRINT" Structuur"TAB(22)"Parameters"
2040 PRINT
2050 FOR I=1 TO E:GOSUB 400:PRINTA(I,1);
2060 AI=A(I,2):PRINTTAB(5);MID$(C$,AI*3-2,3);
2070 IF AI<2 THEN PRINT:GOTO 2140
2080 IF AI<3 THEN PRINTTAB(22);A(I,4):GOTO 2140
2090 IF AI<4 THEN PRINTTAB(22);A(I,3);TAB(11);A(I
,4):GOTO 2140
2100 IF AI=6 THEN PRINTTAB(11);A(I,3);TAB(16);RL(
1);TAB(22);A(I,4):PRINTTAB(11);RL(2);TAB(16);RL(3
):GOTO 2140
2110 IF AI<7 THEN PRINTTAB(11);A(I,3);TAB(22);A(I
,4):GOTO 2140
2120 IF AI<12 THEN PRINTTAB(11);A(I,3):GOTO 2140
2130 PRINTTAB(11);A(I,3);TAB(16);A(I,4)
2140 NEXT
2150 PRINT
2160 PRINT"Output blocks  ";
2170 FOR I=1 TO Q:PRINTTAB(12+I*5);P(I);:NEXT
2180 PRINT
2190 PRINT"Max. Amplitude  :\"U
2200 PRINT"Min. amplitude  :\"R
2210 PRINT"Total time     :\"T1\"sec\"
2220 PRINT"Stapgrootte SD  :\"D1\"sec\"
2230 PRINT"Stapgrootte ST  :\"T2\"sec\"
2240 RETURN
2250 '-----
2260 ' Lees datafile in
2270 NS=-1:PRINT
2280 INPUT"tape or disk (t/d)";A$:GOSUB 1840
2290 IF A$<>"D" AND A$<>"T" THEN 2280
2300 INPUT"Filenaam  : ";F$
2310 IF A$="T" THEN 2340
2320 OPEN F$ FOR INPUT AS #1
2330 GOTO 2350
2340 OPEN "CAS:"+F$ FOR INPUT AS #1
2350 INPUT#1,E,Q,R,U,D1,T1,T2,NP
2360 FOR I=1 TO E:FOR J=1 TO 4
2370 INPUT#1,A(I,J)
2380 NEXT:NEXT
2390 FOR I=1 TO Q:INPUT#1,P(I):NEXT

```

```

2400 FOR I=1 TO NP:INPUT#1,FX(I),FFX(I),RC(I):NEX
T
2410 INPUT#1,RL(1),RL(2),RL(3)
2420 CLOSE #1
2430 RETURN
2440 '-----
2450 ' Schryf datafile weg
2460 PRINT:SCREEN ,,,1
2470 INPUT"tape or disk (t/d)";A$:GOSUB 1840
2480 IF A$<>"T"AND A$<>"D" THEN 2470UB OK00
2490 INPUT"Filenaam: ";F$
2500 IF A$="T" THEN 2530
2510 OPEN F$ FOR OUTPUT AS #1
2520 GOTO 2540
2530 OPEN "cas:"+F$ FOR OUTPUT AS #1
2540 PRINT#1,E,Q,R,U,D1,T1,T2,NP
2550 FOR I=1 TO E:FOR J=1 TO 4
2560 PRINT#1,A(I,J)
2570 NEXT:NEXT
2580 FOR I=1 TO Q:PRINT#1,P(I):NEXT
2590 FOR I=1 TO NP:PRINT#1,FX(I),FFX(I),RC(I):NEX
T
2600 PRINT#1,RL(1),RL(2),RL(3)
2610 CLOSE #1
2620 RETURN
2630 '-----
2640 ' Initialisatie
2650 CLS:T=0:APR=0
2660 ON CNR GOTO 2670,2700,2740,2850
2670 PRINT"TIME";:FOR I=1 TO Q:PRINTTAB(I*9);"BLO
CK";P(I);:NEXT
2680 PRINT:PRINT STRING$(39,45);
2690 RETURN
2700 LPRINT"SIMULATIE-TABEL":LPRINT
2710 LPRINT"TIME";:FOR I=1 TO Q:LPRINTTAB(I*16);"
BLOCK ";P(I);:NEXT
2720 LPRINT:LPRINT STRING$(80,45)
2730 RETURN
2740 Z!=191/(U-R):TC!=255/T1
2750 SCREEN 2
2760 IF U<0 OR R>0 THEN 2830
2770 Y0=191+R*Z
2780 FOR IX=0 TO 10
2790 Y0=191+R*Z
2800 LINE (IX*25.5,Y0)-((IX+1)*25.5,Y0)
2810 LINE -((IX+1)*25.5,Y0-1)

```

```

2820 NEXT IX
2830 LINE (0,0)-(0,191)
2840 RETURN
2850 IF SGN(U)=SGN(R) THEN Z%=0 ELSE Z%=80-U/(U-R
)*79
2860 LPRINT"SIMULATIE-GRAFIEK":LPRINT
2870 FAC=(U-R)/79:LPRINT"Schaal=Hor.: ";D1"sec.  V
er.:"FAC". "
2880 LPRINT STRING$(80,45);
2890 RETURN
2900 '-----
2910 ' Berekeningen en plotten
2920 CNR=(INSTR("STHTSDHD",A$)+1)/2
2930 IF NOT NS THEN 3110 ELSE NS=0
2940 FOR I=1 TO E:B(I)=I:NEXT
2950 J=1:X1=E
2960 H=A(B(J),2):IF H<5 OR H=8 THEN J=J+1:GOTO 29
80
2970 K=B(J):FOR I=J TO X1-1:B(I)=B(I+1):NEXT:B(X1
)=K:X1=X1-1
2980 IF J<=X1 THEN 2960
2990 FOR J=X1+1 TO E
3000 IF X1+1>E GOTO 3070
3010 L=0:K=3:GOSUB 510
3020 IF L=0 THEN 3080
3030 IF A(B(J),2)<12 THEN 3060
3040 L=0:K=4:GOSUB 510
3050 IF L=0 THEN 3080
3060 NEXT
3070 GOTO 3110
3080 G=B(J)
3090 FOR N=J+1 TO E:B(N-1)=B(N):NEXT
3100 B(E)=G:GOTO 3010
3110 GOSUB 2650' initialisatie
3120 FOR J=1 TO E:S(J)=0:NEXT
3130 FOR J=1 TO X1:BJ=B(J)
3140 IF A(BJ,2)<>3 THEN S(A(BJ,1))=A(BJ,4):GOTO 3
160
3150 IF A(BJ,2)=0 THEN S(A(BJ,1))=1 ELSE S(A(BJ,1
))=0
3160 NEXT
3170 'IF CNR=3 THEN SCREEN 2
3180 J=X+1:IF J>E THEN 3220 ELSE 3200
3190 J=1
3200 BJ=B(J):I1=A(BJ,1):I2=A(BJ,2):I3=A(BJ,3):I4=
A(BJ,4)

```

```

3210 ON I2 GOSUB 3680,3690,3700,3710,3720,3730,37
60,3790,3800,3810,3820,3830,3840,3850,3860
3220 IF J<E THEN J=J+1:GOTO 3200
3230 ON CNR GOSUB 3340,3420,3490,3580
3240 T=T+D1:H$=INKEY$:IF T>T1 THEN 3300
3250 IF H$="" THEN 3190
3260 BEEP: SX%=POS(0):SY%=CSRLIN
3265 LOCATE 15,24:PRINT"STOPPEN (J=JA)";
3270 H$=INKEY$:IF H$="" THEN 3270
3280 A%=ASC(H$):IF A%>90 THEN A%=A%-32
3290 IF A%=74 THEN 3310 ELSE LOCATE 15,24:PRINT S
TRING$(18," ");LOCATE SX%,SY%:GOTO 3190
3300 IF LEFT$(A$,1)="S" AND CNR<>3 THEN GOSUB 620
3310 CLS:RETURN
3320 '-----
3330 ' Tabel naar scherm
3340 IF T<APR THEN RETURN
3350 GOSUB 400
3360 APR=APR+T2:PRINT;T;
3370 FOR I=1 TO Q:PRINTTAB(I*9);S(P(I));:NEXT
3380 PRINT:PRINT STRING$(39,45)
3390 RETURN
3400 '-----
3410 ' Tabel naar printer
3420 IF T<APR THEN RETURN
3430 APR=APR+T2:LPRINT;T;
3440 FOR I=1 TO Q:LPRINTTAB(I*16);S(P(I));:NEXT:L
PRINT
3450 LPRINT STRING$(80,45)
3460 RETURN
3470 '-----
3480 ' Grafiek naar scherm
3490 FOR I=1 TO Q:SP!=S(P(I))
3500 IF SP!>U THEN SP!=U
3510 IF SP!<R THEN SP!=R
3520 PSET(T*TC!,191-((SP!-R)*Z!)),15
3530 NEXT I
3540 IF T+D1>T1 THEN GOSUB 4100:RETURN
3550 RETURN
3560 '-----
3570 ' Grafiek naar printer
3580 RESTORE:READ A,B:P$=STRING$(80,32)
3590 IF Z%<>0 THEN P$=LEFT$(P$,Z%-1)+"I"+RIGHT$(P
$,80-Z%)
3600 FOR I=1 TO Q:READ C:SPI%=(S(P(I))-R)/FAC+1
3610 IF SPI%>80 OR SPI%<1 THEN SPI%=1-79*(SPI%>80
)

```

```

3620 P$=LEFT$(P$,SPI%-1)+CHR$(C)+RIGHT$(P$,80-SPI
%)
3630 NEXT
3640 DATA 248,15,42,35,43,37
3650 LPRINT P$;:RETURN
3660 '-----
3670 ' functieblokken
3680 S(I1)=T:RETURN
3690 RETURN
3700 S(I1)=1+((T>I4)OR(T<I3)):RETURN
3710 S(I1)=S(I1)+S(I3)*D1:RETURN
3720 S(I1)=S(I3)*I4:RETURN
3730 NR=2:IF ABS(S(I3)-I4)<1E-04 THEN 3750
3740 NR=(S(I3)-I4>0)*-2+1
3750 S(I1)=S(RL(NR)):RETURN
3760 N=1:H=S(I3)
3770 N=N+1:IF (FX(N)<=H) AND (N<NP) THEN 3770
3780 S(I1)=FFX(N)-(FX(N)-H)*RC(N):RETURN
3790 S(I1)=I4:A(B(J),4)=S(I3):RETURN
3800 S(I1)=SIN(S(I3)):RETURN
3810 S(I1)=COS(S(I3)):RETURN
3820 S(I1)=EXP(S(I3)):RETURN
3830 S(I1)=S(I3)+S(I4):RETURN
3840 S(I1)=S(I3)-S(I4):RETURN
3850 S(I1)=S(I3)*S(I4):RETURN
3860 S(I1)=S(I3)/S(I4):RETURN
3870 '-----
3880 ' FNC en REL routine
3890 IF K=7 THEN 3960' FNC
3900 T=0
3910 T=T+1:I=J+1:BAD=(I>LEN(A$)):GOSUB 460
3920 RL(T)=VAL(MID$(A$,I,J-I))
3930 IF T<3 AND (NOT BAD) THEN 3910
3940 IF BAD THEN 910 ELSE 890
3950 ' FNC-blok (=function)
3960 NP=A(M,4)
3970 INPUT"x, F(x)=";FX(1),FFX(1)
3980 FOR I=2 TO NP
3990 INPUT"x, F(x)=";FX(I),FFX(I)
4000 RC(I)=(FFX(I)-FFX(I-1))/(FX(I)-FX(I-1))
4010 NEXT I
4020 PRINT:GOTO 890
4030 '-----
4040 ' foutbehandeling
4050 IF ERR=6 THEN CLS:PRINT"Overflow in block"11
:PRINT:RESUME 240

```

```
4060 IF ERR=11 THEN CLS:PRINT"division by zero":R
ESUME 240
4070 CLS:PRINT"Error";ERR:RESUME 240
4080 '-----
4090 ' einde schermplot
4100 BEEP:BEEP:PSET (240,177)
4110 OPEN "grp:" AS #1
4120 PRINT #1,"?"
4130 I$=INKEY$
4140 IF INKEY$<>CHR$(13) THEN 4140
4150 CLOSE #1:RETURN
```


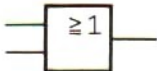
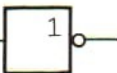

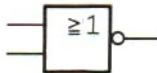
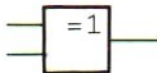




### 5.3 DIGITALE SIMULATIE

Het digitale simulatieprogramma is speciaal bedoeld voor het doorrekenen van combinatorische schakelingen. Een combinatorische schakeling is niet tijdsafhankelijk en bevat zodoende geen terugkoppelingen. Bij iedere ingangscombinatie hoort slechts één uitgangscombinatie. Dit is tegenstelling tot zogenaamde sequentiële schakelingen en clock-schakelingen.

#### 5.3.1 DE BOUWSTENEN

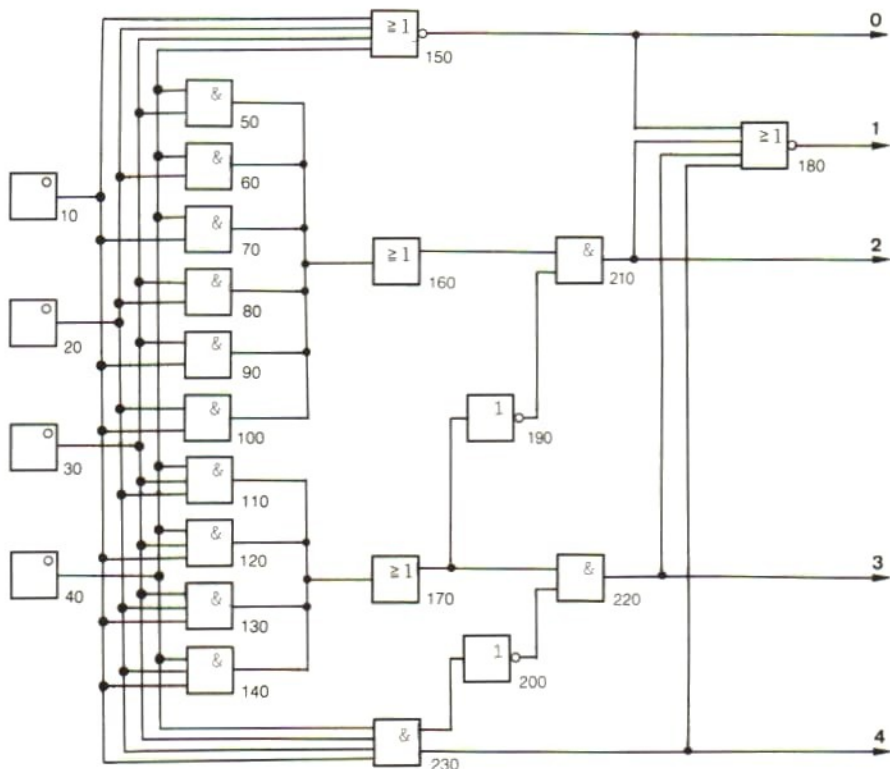
De te simuleren schakeling dient te worden opgebouwd uit de volgende logische elementen:

		aantal toegestane ingangen
AND		1 - M
OR		1 - M
INV		1
NAND		1 - M
NOR		1 - M
EXOR		2
NEXOR		2
ING		0

Het maximaal toegestane ingangen is M. De waarde van M wordt door het programma gevraagd (zie volgende paragraaf).

#### 5.3.2 EEN UITGEWERKT VOORBEELD

Om een en ander toe te lichten gebruiken we als voorbeeld een zogenaamde ingangenteller. De schakeling heeft 4 ingangen en 5 uitgangen.



Uitgang 1 heeft als waarde 1 als alle ingangen de waarde 0 hebben. Uitgang 2 heeft de waarde 1 als één van de ingangen de waarde 1 heeft. Uitgang 3 heeft de waarde 1 als twee van de ingangen de waarde 1 hebben, enz.

Om de structuur van de schakeling te kunnen invoeren, geven we ieder element een uniek nummer. Omdat alle elementen slechts één uitgang hebben, wordt dit nummer tevens gebruikt als aanduiding van de uitgang van het element. Een element wordt nu als volgt ingevoerd:

bloknummer; type; ingangen

Blok 70 bijvoorbeeld is een AND en de ingangen zijn verbonden met de uitgang van blokken 10 en 40. Zo wordt blok 70 als volgt aangegeven:

70; AND; 10; 40

Op deze wijze voeren we de gehele schakeling in.

### 5.3.3 INVOEREN VAN EEN MODEL

Na opstarten van het programma verschijnt het volgende menu:

---Invoer model---

Invoer van:

C: Cassette  
D: Diskette  
T: Toetsenbord  
G: Geen

Omdat we een nieuw model willen invoeren, kiezen we voor de T. Met de keuze C of D kunnen we een model laden van respectievelijk cassette of diskette. Met de keuze G blijft een eventueel reeds in het geheugen aanwezige model onveranderd.

Om zo veel mogelijk geheugen te besparen vraagt het programma vervolgens naar het maximaal aantal ingangen per element. In ons voorbeeld heeft blok 160 de meeste ingangen, namelijk 6. We geven dus 6 op.

Daarna verschijnt een dubbele punt (:). We kunnen nu de schakeling invoeren volgens de in de vorige paragraaf beschreven notatie. Ieder blok moet afgesloten worden met een <RETURN>.

Voor het voorbeeld ziet de invoer er als volgt uit:

---Model invoeren---

Maximaal aantal ingangen per element: ?

6

```
: ? 10; ING
: ? 20; ING
: ? 30; ING
: ? 40; ING
: ? 50; AND; 30; 40
: ? 60; AND; 20; 40
: ? 70; AND; 10; 40
: ? 80; AND; 20; 30
: ? 90; AND; 10; 30
: ? 100; AND; 10; 20
: ? 110; AND; 20; 30; 40
: ? 120; AND; 10; 30; 40
: ? 130; AND; 10; 20; 30
: ? 140; AND; 10; 20; 40
: ? 150; NOR; 10; 20; 30; 40
: ? 160; OR; 50; 60; 70; 80; 90; 100
: ? 170; OR; 110; 120; 130; 140
: ? 230; AND; 10; 20; 30; 40
: ? 190; INV; 170
: ? 200; INV; 230
: ? 210; AND; 160; 190
: ? 220; AND; 170; 200
: ? 180; NOR; 150; 210; 220; 230
: ?
```

De volgorde waarin de blokken worden ingevoerd is niet van belang. Een eventueel foutief ingevoerd blok kan gewoon nogmaals worden ingevoerd. Een reeds ingevoerd blok kunnen we weer verwijderen door alleen het bloknummer in te toetsen, gevolgd door <RETURN>.

De invoer wordt beëindigd door alleen <RETURN> te geven. Onmiddellijk verschijnt de prompt (#).

#### 5.3.4 DE COMMANDO'S

Het verschijnen van de prompt (#) geeft aan dat we nu één van de volgende commando's kunnen geven:

---Menu---

```
S : Start simulatie
WT: Opstellen waarheidstabel
L : Laten zien van model
C : Laten zien van schijfcatalogus
NM: Nieuw model invoeren
VM: Veranderen model
VI: Veranderen ingangen en
    start simulatie
DU: Definiëren uitgangen
PU: Uitvoer laten zien
OD: Opslaan van model op diskette
OC: Opslaan van model op cassette
H : Deze commandolijst laten zien
ST: Verlaten van het programma
```

# ?

We zullen de commando's één voor één nader toelichten:

##### DU: Definiëren uitgangen

Met behulp van het DU-commando geven we op van welke blokken we de uitgangswaarde willen weten, m.a.w. welk blok een uitgangsblok is.

Na het intoetsen van het commando verschijnen één voor één de nummers van de blokken die reeds als uitgangsblok zijn gedefinieerd. Door het intoetsen van de letter N gevolgd door <RETURN>, geven we aan dat het desbetreffende blok niet meer als uitgangsblok dient te worden beschouwd. Typen we alleen een <RETURN> in, dan blijft het blok een uitgangsblok. Als alle uitgangsblokken aan de beurt zijn geweest, kunnen we nieuwe uitgangsblokken toevoegen.

Aangezien we in ons voorbeeld tot nu toe nog geen enkel blok als uitgangsblok hadden gedefinieerd en we de blokken 150, 180, 210, 220, en 230 als uitgang willen definiëren, krijgen we de volgende invoer:

---Definiëren uitgangen---

oude uitgangen:  
(Toets een N in voor wissen)

Nieuwe uitgangen:

: ? 150  
: ? 180  
: ? 210  
: ? 220  
: ? 230  
: ?

L: Laten zien van model

Het L-commando geeft een listing van het ingevoerde model. Voor ons voorbeeld:

---Listing van model---

Maximaal aantal ingangen voor AND, OR, N  
AND en NOR: 6

No:	Soort:	Uit:	In:
10	ING	0	
20	ING	0	
30	ING	0	
40	ING	0	
50	AND	0	30 ; 40
60	AND	0	20 ; 40
70	AND	0	10 ; 40
80	AND	0	20 ; 30
90	AND	0	10 ; 30
100	AND	0	10 ; 20
110	AND	0	20 ; 30 ; 40
120	AND	0	10 ; 30 ; 40
130	AND	0	10 ; 20 ; 30
140	AND	0	10 ; 20 ; 40
U 150	NOR	0	10 ; 20 ; 30 ; 40
160	OR	0	50 ; 60 ; 70 ; 80 ;
90 ; 100			
170	OR	0	110 ; 120 ; 130 ; 140
U 180	NOR	0	150 ; 210 ; 220 ; 230
190	INV	0	170
200	INV	0	230
U 210	AND	0	160 ; 190
U 220	AND	0	170 ; 200
U 230	AND	0	10 ; 20 ; 30 ; 40

De listing is gesorteerd op opklimmend bloknummer. Bij sommige blokken staat in de eerste kolom een "U". Dit geeft aan dat het betreffende blok een uitgangsblok is. Mocht blijken dat we fouten hebben gemaakt met het invoeren van het model, dan kunnen we het wijzigen met het VM-commando.

#### VM: Veranderen model

Voor het veranderen van het model gaan we op precies dezelfde manier te werk als bij het invoeren van het model. Weer verschijnt de dubbele punt (:). We kunnen nu nieuwe blokken toevoegen of bestaande blokken verwijderen. Beëindig het commando door alleen een <RETURN> te geven.

#### S : Start simulatie

Als we het model correct hebben ingevoerd en de uitgangsblokken hebben gedefinieerd, kunnen we de schakeling gaan simuleren met het S-commando. Het gehele model wordt nu doorgerekend en na enige tijd verschijnt:

---Resultaten---

Ingangen:	Uitgangen:
1Ø : Ø	15Ø ; 1
2Ø : Ø	18Ø ; Ø
3Ø : Ø	21Ø ; Ø
4Ø : Ø	22Ø ; Ø
	23Ø ; Ø

Links zijn de ingangsblokken aangegeven met hun ingangswaarde en rechts staan de uitgangsblokken met hun uitgangswaarde. Omdat alle ingangen de waarde 0 hebben, heeft alleen uitgangsblok 1 de waarde 1.

#### VI: Veranderen ingangen en start simulatie

Uiteraard willen we ook voor andere ingangscombinaties weten wat de waarde van de uitgangsblokken is. Door middel van het VI-commando kunnen we de ingangen een nieuwe waarde geven. Op het beeldscherm verschijnt:

---Veranderen ingangen---

```
ING 1Ø ( Ø ) : ?  
ING 2Ø ( Ø ) : ? 1  
ING 3Ø ( Ø ) : ? 1  
ING 4Ø ( Ø ) : ?
```

Even geduld a.u.b

---Resultaten---

Ingangen:	Uitgangen:
10 : 0	150 ; 0
20 : 1	180 ; 0
30 : 1	210 ; 1
40 : 0	220 ; 0
	230 ; 0

Voor iedere ingang toetsen we een 1 of een 0 in, gevolgd door <RETURN>. Drukken we alleen op <RETURN>, dan blijft de ingangswaarde onveranderd.

Na veranderen van de laatste ingangswaarde vindt automatisch een simulatie plaats. Na enige tijd worden de nieuwe uitgangswaarden weergegeven.

**WT: Opstellen van de waarheidstabel**

In plaats van zelf steeds de ingangswaarden te veranderen met het VI-commando, kunnen we het model laten doorrekenen voor alle mogelijke ingangscombinaties met behulp van het WT-commando.

We verkrijgen dan een zogenaamde waarheidstabel. Voor ons voorbeeld:

Het opstellen zal ongeveer 6 min.  
duren.

Start? (J/N): ? J

	Ingangen:	Uitgangen:
1	0 0 0 0	1 0 0 0 0
2	0 0 0 1	0 1 0 0 0
3	0 0 1 0	0 1 0 0 0
4	0 0 1 1	0 0 1 0 0
5	0 1 0 0	0 1 0 0 0
6	0 1 0 1	0 0 1 0 0
7	0 1 1 0	0 0 1 0 0
8	0 1 1 1	0 0 0 1 0
9	1 0 0 0	0 1 0 0 0
10	1 0 0 1	0 0 1 0 0
11	1 0 1 0	0 0 1 0 0
12	1 0 1 1	0 0 0 1 0
13	1 1 0 0	0 0 1 0 0
14	1 1 0 1	0 0 0 1 0
15	1 1 1 0	0 0 0 1 0
16	1 1 1 1	0 0 0 0 1

# ?

Omdat het aantal ingangscombinaties groot is, duurt het samenstellen van de waarheidstabel enige tijd. Het is ook mogelijk zelf de ingangscombinaties op te geven waarvoor de waarheidstabel moet worden berekend. Daartoe beantwoorden we de vraag "Wilt u zelf de gewenste ingangscombinaties opgeven?" met "J".

We mogen maximaal 20 combinaties opgeven, bijvoorbeeld:

```
0; 0; 0; 0; <RETURN>
0; 0; 0; 1 <RETURN>
0; 0; 1; 1 <RETURN>
0; 1; 1; 1 <RETURN>
1; 1; 1; 1 <RETURN>
<RETURN>
```

We verkrijgen het volgende resultaat:

No:	Ingangen:	Uitgangen:
1	Ø Ø Ø Ø	1 Ø Ø Ø Ø
2	Ø Ø Ø 1	Ø 1 Ø Ø Ø
3	Ø Ø 1 1	Ø Ø 1 Ø Ø
4	Ø 1 1 1	Ø Ø Ø 1 Ø
5	1 1 1 1	Ø Ø Ø Ø 1

**OD:** Opslaan van model op diskette

**OC:** Opslaan van model op cassette

Met de commando's **OD** en **OC** kunnen we een model saveen op diskette ef cassette. Het model kan weer worden geladen met het **NM**-commando.

**NM:** Nieuw model invoeren

Na intoetsen van het **NM**-commando verschijnt hetzelfde menu als bij het opstarten van het programma. We hebben de keus om een model van cassette (**C**) of diskette(**D**) te laden, óf we kunnen een nieuw model invoeren (**T**).

**PU:** Uitvoer laten zien

De laatst berekende uitvoer wordt nogmaals getoond.

**H :** Deze commandolijst laten zien.

**ST:** Verlaten van programma.



Programma:

```
10 REM -----
20 REM ---          ---
30 REM ---      Digisim      ---
40 REM ---          ---
50 REM ---      door:      ---
60 REM ---      Chris Baten  ---
70 REM ---          ---
80 REM -----
90 CLEAR 200
100 ON ERROR GOTO 4080
110 WIDTH 40
120 DEFINT A-Z.
130 DIM IC(1), IN(1), ST(1,1)
140 CLS
150 LOCATE 16,6
160 PRINT "DIGISIM"
170 LOCATE 6,8
180 PRINT "Een simulatieprogramma voor"
190 LOCATE 12:PRINT "Combinatorische"
200 LOCATE 9:PRINT "digitale schakelingen"
210 PRINT
220 LOCATE 18:PRINT "door:"
230 PRINT
240 LOCATE 14:PRINT "Chris Baten"
250 FOR I=1 TO 1500
260 NEXT I
270 CLS
280 LOCATE 11
290 PRINT "----Invoer model----"
300 LOCATE 8,4
310 PRINT "Invoer van: "
320 LOCATE 13,6
330 PRINT "C: Cassette"
340 LOCATE 13
350 PRINT "D: Diskette"
360 LOCATE 13
370 PRINT "T: Toetsenbord"
380 LOCATE 13
390 PRINT "G: Geen"
400 LOCATE ,15
410 INPUT ": ";A$
420 IF A$="C" THEN GOSUB 490:GOTO 470
430 IF A$="D" THEN GOSUB 730:GOTO 470
440 IF A$="T" THEN GOTO 1360
450 IF A$="G" THEN GOTO 470
460 GOTO 410
470 GOTO 5410
480 END
490 REM ---Invoer van cassette---
500 CLS:PRINT:ERASE IC, IN, ST
510 DIM IC (20,16), IN(16), ST(1,1)
520 LOCATE 8:PRINT "----Invoer van cassette----"
530 LOCATE 7,3
```

```

540 INPUT "Naam van het model: ";NA$
550 LOCATE 10,5
560 PRINT "Start de recorder"
570 LOCATE 10:PRINT "(PLAY-stand)"
580 LOCATE 10:PRINT "en druk een toets in."
590 A$=INKEY$:IF A$="" THEN GOTO 590
600 LOCATE 8,10
610 PRINT "Het model wordt geladen"
620 NA$="CAS:"+NA$
630 OPEN NA$ FOR INPUT AS #1
640 INPUT #1,TEL,ME,MI
650 ERASE ST:DIM ST(ME,MI+4)
660 FOR I=0 TO ME
670 FOR J=0 TO MI+4
680 INPUT #1,ST(I,J)
690 NEXT J
700 NEXT I
710 CLOSE
720 RETURN
730 REM ---Invoer van diskette---
740 CLS
750 LOCATE 8,1
760 PRINT "---Invoer van diskette---"
770 PRINT
780 ON ERROR GOTO 4080
790 ERASE ST,IN,IC
800 DIM IC(20,16),IN(16),ST(1,1)
810 C=1
820 PRINT
830 NA$=""
840 INPUT "Naam van het model: ";NA$
850 PRINT
860 IF NA$="" THEN GOTO 5780
870 NA$="A:"+NA$
880 OPEN NA$ FOR INPUT AS #1
890 INPUT #1,TEL,MI,ME
900 ERASE ST
910 DIM ST(ME,MI+4)
920 FOR I=0 TO ME
930 FOR J=0 TO MI+4
940 INPUT #1,ST(I,J)
950 NEXT J
960 NEXT I
970 CLOSE
980 GOTO 5780
990 RETURN
1000 REM ---Uitvoer naar cassette---
1010 CLS
1020 LOCATE 7
1030 PRINT "---Uitvoer naar cassette---"
1040 LOCATE ,10
1050 INPUT "Naam van het te saven model: ";NA$
1060 NA$="CAS:"+NA$
1070 OPEN NA$ FOR OUTPUT AS #1
1080 PRINT #1,TEL,ME,MI
1090 FOR I=0 TO ME
1100 FOR J=0 TO MI+4

```

```

1110 PRINT #1,ST(I,J)
1120 NEXT J
1130 NEXT I
1140 CLOSE
1150 RETURN
1160 REM ---Uitvoer naar diskette---
1170 CLS
1180 PRINT
1190 LOCATE 8
1200 PRINT "---Uitvoer naar diskette---"
1210 LOCATE 1,5
1220 NA$=""
1230 INPUT "Naam van model: ";NA$
1240 IF NA$="" THEN GOTO 5780
1250 NA$="A:"+NA$
1260 OPEN NA$          FOR OUTPUT AS #1
1270 PRINT #1,TEL,MI,ME
1280 FOR I=0 TO ME
1290 FOR J=0 TO MI+4
1300 PRINT #1,ST(I,J)
1310 NEXT J
1320 NEXT I
1330 CLOSE
1340 PRINT
1350 RETURN
1360 REM ---Invoer toetsenbord---
1370 ERASE IC,ST,IN
1380 DIM IC(20,16),IN(16)
1390 CLS
1400 C=0
1410 LOCATE 12
1420 PRINT "---Model invoeren---"
1430 LOCATE ,4
1440 INPUT "Maximaal aantal ingangen per element: ";MI
1450 PRINT
1460 ME=INT(1000/(MI+4))
1470 DIM ST(ME,MI+4)
1480 TEL=0
1490 ST$=""
1500 INPUT ": ";ST$
1510 IF ST$="" THEN PRINT :GOTO 5600
1520 S$=LEFT$(ST$,1)
1530 GOSUB 4040:IF F=1 THEN 1490
1540 N$=S$
1550 FOR I=2 TO 5
1560 S$=MID$(ST$,I,1)
1570 IF S$=";" THEN GOTO 1620
1580 IF S$="" THEN GOSUB 4620 :GOTO 1490
1590 GOSUB 4040:IF F=1 THEN 1490
1600 N$=N$+S$
1610 NEXT I
1620 ZOEK=VAL(N$)
1630 GOSUB 4120
1640 IF G=0 THEN TEL=TEL+1:PS=TEL:GOTO 1670
1650 IF G=1 THEN GOTO 1670
1660 IF G=2 THEN TEL=TEL+1:GOSUB 4730

```

```

1670 IF TEL>=ME THEN PRINT "Model is te groot.":PRINT "Verdee
l het model in stukken of pas de declaratie van het modelarra
y ST aan."
1680 ST(PS,0)=0
1690 ST(PS,1)=VAL(N$)
1700 K=I
1710 N$=""
1720 FOR I=K+1 TO K+5
1730 S$=MID$(ST$,I,1)
1740 IF S$=";" THEN GOTO 1770
1750 N$=N$+S$
1760 NEXT I
1770 IF N$="AND" THEN ST(PS,2)=1:GOTO 1860
1780 IF N$="OR" THEN ST(PS,2)=2:GOTO 1860
1790 IF N$="NAND" THEN ST(PS,2)=3:GOTO 1860
1800 IF N$="NOR" THEN ST(PS,2)=4:GOTO 1860
1810 IF N$="EXOR" THEN ST(PS,2)=5:GOTO 1860
1820 IF N$="NEXOR" THEN ST(PS,2)=6:GOTO 1860
1830 IF N$="INV" THEN ST(PS,2)=7:GOTO 1860
1840 IF N$="ING" THEN ST(PS,2)=8:GOTO 1860
1850 GOSUB3990:GOTO 1490
1860 S$=MID$(ST$,I+1,1)
1870 AANTAL=0
1880 IF S$="" THEN GOTO 2070
1890 GOSUB 4040:IF F=1 THEN 1490
1900 I$=S$
1910 K=I
1920 FOR I=K+2 TO LEN(ST$)+1
1930 S$=MID$(ST$,I,1)
1940 IF S$=";" OR S$="" THEN GOTO 1980
1950 GOSUB 4040:IF F=1 THEN 1490
1960 I$=I$+S$
1970 GOTO 2060
1980 AANTAL=AANTAL+1
1990 IF AANTAL>MI THEN GOTO 2190
2000 ST(PS,AANTAL+4)=VAL(I$)
2010 IF S$="" THEN GOTO 2070
2020 I=I+1
2030 S$=MID$(ST$,I,1)
2040 GOSUB 4040:IF F=1 THEN 1490
2050 I$=S$
2060 NEXT I
2070 ST(PS,4)=AANTAL
2080 IF ST(PS,2)>4 THEN GOTO 2110
2090 IF AANTAL<1 THEN GOTO 2190
2100 GOTO 2180
2110 IF ST(PS,2)>6 THEN GOTO 2140
2120 IF AANTAL<>2 THEN GOTO 2190
2130 GOTO 2180
2140 IF ST(PS,2)<>7 THEN GOTO 2170
2150 IF AANTAL<>1 THEN GOTO 2190
2160 GOTO 2180
2170 IF AANTAL<>0 THEN GOTO 2190
2180 GOTO 1490
2190 PRINT "Niet toegestaan aantal ingangen."
2200 N$=STR$(ST(PS,1))
2210 GOSUB 4620

```

```

2220 GOTO 1490
2230 REM ---And---
2240 UIT=1
2250 FOR S=1 TO ST(NW,4)
2260 ZOEK=ST(NW,S+4)
2270 GOSUB 4120
2280 IF G<>1 THEN GOTO 3930
2290 IF ST(PS,3)=0 THEN UIT =0:GOTO 2310
2300 NEXT S
2310 ST(NW,3)=UIT
2320 RETURN
2330 REM ---OR---
2340 UIT=0
2350 FOR S=1 TO ST(NW,4)
2360 ZOEK=ST(NW,S+4)
2370 GOSUB 4120
2380 IF G<>1 THEN GOTO 3930
2390 IF ST(PS,3)=1 THEN UIT=1:GOTO 2410
2400 NEXT S
2410 ST(NW,3)=UIT
2420 RETURN
2430 REM ---Nand---
2440 GOSUB 2230:REM ---And---
2450 GOSUB 2740:REM ---InvZ---
2460 RETURN
2470 REM ---Nor---
2480 GOSUB 2330:REM ---Or---
2490 GOSUB 2740:REM ---InvZ---
2500 RETURN
2510 REM ---Exor---
2520 ZOEK=ST(NW,5)
2530 GOSUB 4120:REM ---Zoek element uit lijst---
2540 IF G<>1 THEN GOTO 3930
2550 P1=PS
2560 ZOEK=ST(NW,6)
2570 GOSUB 4120:REM ---Zoek element uit lijst---
2580 IF G<>1 THEN GOTO 3930
2590 P2=PS
2600 UIT=0
2610 IF ST(P1,3)<>ST(P2,3) THEN UIT=1
2620 ST(NW,3)=UIT
2630 RETURN
2640 REM ---Nexor--
2650 GOSUB 2510:REM ---Exor---
2660 GOSUB 2740:REM ---InvZ---
2670 RETURN
2680 REM ---Inv---
2690 ZOEK=ST(NW,5)
2700 GOSUB 4120
2710 IF G<>1 THEN GOTO 3930
2720 ST(NW,3)=1-ST(PS,3)
2730 RETURN
2740 REM ---InvZ---
2750 ST(NW,3)=1-ST(NW,3)
2760 RETURN
2770 REM ---Simulatie---
2780 K=1

```

```

2790 V=0
2800 IF WT=1 THEN GOTO 2830
2810 CLS:LOCATE 10,12
2820 PRINT "Even geduld a.u.b...."
2830 FOR NW=1 TO TEL
2840 H=ST(NW,3)
2850 ON ST(NW,2) GOSUB 2230,2330,2430,2470,2510,2640,2680
2860 IF H<>ST(NW,3) THEN V=1
2870 NEXT NW
2880 K=K+1
2890 IF K>30 THEN PRINT "Er treedt oscillatie op. De simulati
e":PRINT "is afgebroken na ";K-1;" maal doorrekenen.": GOTO 2
930
2900 IF V=1 THEN GOTO 2790
2910 IF WT=1 THEN GOTO 2930
2920 GOSUB 5270:REM ---Uitvoer---
2930 RETURN
2940 REM ---Opstellen waarheidstabel---
2950 CLS:LOCATE 6
2960 PRINT "---Opstellen waarheidstabel---"
2970 PRINT
2980 INPUT "Wilt U zelf de ingangenreeks definiëren? (J/N) ";
A$
2990 IF A$="j" OR A$="J" THEN GOTO 3450
3000 IN(1)=-1
3010 FOR I=2 TO 16
3020 IN(I)=0
3030 NEXT I
3040 AI=0
3050 FOR T=1 TO TEL
3060 IF ST(T,2)=8 THEN AI=AI+1
3070 IF AI>=16 THEN PRINT "Te groot aantal ingangen.": GOTO 3
440
3080 NEXT T
3090 LOCATE ,4
3100 PRINT "Het opstellen zal ongeveer ";INT(TEL*2^AI/60);" m
in.          ":LOCATE ,5:PRINT "duren.          "
3110 INPUT "Start? (J/N): ";A$
3120 IF A$="n" OR A$="N" THEN GOTO 3440
3130 LOCATE 6,8
3140 PRINT "Ingangen:";
3150 LOCATE (AI*2+14)
3160 PRINT "Uitgangen:"
3170 PRINT
3180 FOR R=1 TO 2^AI
3190 FOR I=1 TO AI
3200 IN(I)=IN(I)+1
3210 IF IN(I)<>2 THEN GOTO 3240
3220 IN(I)=0
3230 NEXT I
3240 WT=1
3250 I=1
3260 FOR T=TEL TO 1 STEP -1
3270 IF ST(T,2)=8 THEN ST(T,3)=IN(I):I=I+1
3280 NEXT T
3290 GOSUB 2770
3300 PRINT R;

```

```

3310 LOCATE 6
3320 FOR I=AI TO 1 STEP -1
3330 PRINT USING"##";IN(I);
3340 NEXT I
3350 LOCATE (AI*2+14)
3360 FOR T=1 TO TEL
3370 IF ST(T,0)=1 THEN PRINT USING"##";ST(T,3);
3380 NEXT T
3390 PRINT
3400 IF INT(R/20)*20=R THEN INPUT "Scroll?: ";A$
3410 NEXT R
3420 PRINT
3430 WT=0
3440 RETURN
3450 REM ---DEFINIEREN INGANGENREEKS EN OPSTELLEN WAARHEIDSTA
BEL---
3460 CLS:LOCATE 5: PRINT "---Definiëren ingangenreeks---": PR
INT
3470 PRINT "Geef de gewenste ingangstoestanden: ":PRINT
3480 AI=0
3490 FOR T=1 TO TEL
3500 IF ST(T,2)=8 THEN AI=AI+1
3510 IF AI>16 THEN PRINT: PRINT "Te groot aantal ingangen.":
GOTO 3920
3520 NEXT T
3530 FOR AS=1 TO 20
3540 PRINT AS;
3550 ST$=""
3560 INPUT " : ";ST$
3570 IF ST$="" THEN GOTO 3720
3580 IF LEN(ST$)<>2*AI-1 THEN GOTO 3680
3590 FOR S=1 TO 2*AI-1 STEP 2
3600 S$=MID$(ST$,S,1)
3610 IF S$<>"0" AND S$<>"1" THEN GOTO 3680
3620 IC(AS,(S+1)/2)=VAL(S$)
3630 S$=MID$(ST$,S+1,1)
3640 IF S$="" THEN GOTO 3710
3650 IF S$<>";" THEN GOTO 3680
3660 NEXT S
3670 GOTO 3630
3680 PRINT "Foute invoer."
3690 PRINT "Probeer opnieuw..."
3700 GOTO 3550
3710 NEXT AS
3720 CLS:PRINT " No: ";TAB(6);"Ingangen: ";TAB(AI*2+14);"Uitgan
gen: ":PRINT
3730 FOR I=1 TO AS-1
3740 Z=1
3750 FOR T=1 TO TEL
3760 IF ST(T,2)=8 THEN ST(T,3)=IC(I,Z):Z=Z+1
3770 NEXT T
3780 WT=1: GOSUB 2770: WT=0
3790 PRINT " ";I;
3800 LOCATE 7
3810 IF AI=0 THEN GOTO 3850
3820 FOR J=1 TO AI
3830 PRINT USING"##";IC(I,J);

```

```

3840 NEXT J
3850 LOCATE (AI*2+15)
3860 FOR T=1 TO TEL
3870 IF ST(T,0)=1 THEN PRINT USING"##";ST(T,3);
3880 NEXT T
3890 PRINT
3900 NEXT I
3910 PRINT
3920 RETURN
3930 REM ---Foutmelding onbekende ingang---
3940 PRINT "Struktuurfout"
3950 PRINT "Het blok "ST(NW,1);" heeft de onbekende ingang ";
ZOEK
3960 PRINT
3970 GOTO 5780
3980 RETURN
3990 REM ---Foutmelding invoer---
4000 PRINT "Foute invoer."
4010 PRINT "Probeer opnieuw..."
4020 F=1
4030 RETURN
4040 REM ---Test op cijfer---
4050 F=0
4060 IF ASC(S$)<48 OR ASC(S$)>58 THEN GOTO 3990
4070 RETURN
4080 REM ---Errorbehandeling---
4090 IF ERR=19 THEN PRINT "I/O-error":RESUME 5600
4100 IF ERR=53 OR ERR=56 THEN PRINT "Verkeerde naam voor mode
1.":RESUME 5600
4110 RESUME 5410
4120 REM ---Zoek element in lijst---
4130 G=0
4140 IF TEL=0 THEN GOTO 4190
4150 FOR PS=1 TO TEL
4160 IF ST(PS,1)=ZOEK THEN G=1: GOTO 4190
4170 IF ST(PS,1)>ZOEK THEN G=2: GOTO 4190
4180 NEXT PS
4190 RETURN
4200 REM ---Listing van model---
4210 CLS
4220 LOCATE 7
4230 PRINT "---Listing van model---"
4240 PRINT
4250 PRINT "Maximaal aantal ingangen voor AND, OR, NAND en NO
R: ";MI
4260 PRINT
4270 PRINT "    No: Soort: Uit:    In:"
4280 PRINT
4290 LOCATE,7
4300 IF TEL=0 THEN GOTO 4360
4310 FOR T=1 TO TEL
4320 GOSUB 4370
4330 IF INT(T/16)*16=T THEN INPUT "Scroll?: ";A$
4340 NEXT T
4350 PRINT
4360 RETURN

```



```

4370 REM ---Afdrukken element---
4380 IF ST(T,0)=1 THEN PRINT "U";
4390 LOCATE 2
4400 PRINT ST(T,1);
4410 LOCATE 7
4420 ON ST(T,2) GOTO 4430,4440,4450,4460,4470,4480,4490,4500
4430 PRINT "AND"; : GOTO 4510
4440 PRINT "OR"; : GOTO 4510
4450 PRINT "NAND";: GOTO 4510
4460 PRINT "NOR"; : GOTO 4510
4470 PRINT "EXOR";: GOTO 4510
4480 PRINT "NEXOR";:GOTO 4510
4490 PRINT "INV"; : GOTO 4510
4500 PRINT "ING"; : GOTO 4510
4510 LOCATE 14
4520 PRINT ST(T,3);
4530 LOCATE 18
4540 IF ST(T,4)=0 THEN GOTO 4600
4550 PRINT ST(T,5);
4560 IF ST(T,4)=1 THEN GOTO 4600
4570 FOR R=2 TO ST(T,4)
4580 PRINT ";";ST(T,4+R);
4590 NEXT R
4600 PRINT
4610 RETURN
4620 REM ---Wissen in model---
4630 ZOEK=VAL(N$)
4640 GOSUB 4120:REM ---Zoek element in lijst---
4650 IF G<>1 THEN GOTO 4720
4660 FOR T= PS+1 TO TEL
4670 FOR S=0 TO MI+4
4680 ST(T-1,S)=ST(T,S)
4690 NEXT S
4700 NEXT T
4710 TEL=TEL-1
4720 RETURN
4730 REM ---Schuiven---
4740 FOR T=TEL TO PS+1 STEP -1
4750 FOR S=0 TO MI+4
4760 ST(T,S)=ST(T-1,S)
4770 NEXT S
4780 NEXT T
4790 RETURN
4800 REM ---Veranderen ingangen---
4810 CLS
4820 LOCATE 8,1
4830 PRINT "---Veranderen ingangen---"
4840 PRINT
4850 FOR T=1 TO TEL
4860 LOCATE 11
4870 IF ST(T,2)<>8 THEN GOTO 4950
4880 PRINT "ING ";ST(T,1);" (" ;ST(T,3);") ";
4890 IN$=""
4900 INPUT": ";IN$
4910 IF IN$="" THEN GOTO 4950
4920 IN=VAL(IN$)
4930 IF IN<>0 AND IN<>1 THEN GOTO 4860

```

```

4940 ST(T,3)=IN
4950 NEXT T
4960 PRINT
4970 RETURN
4980 REM ---Definiëren uitgangen---
4990 CLS
5000 LOCATE 7
5010 PRINT "----Definiëren uitgangen----"
5020 LOCATE,4
5030 PRINT "oude uitgangen:"
5040 PRINT "(Toets een N in voor wissen)"
5050 PRINT
5060 FOR T=1 TO TEL
5070 IF ST(T,0)<>1 THEN GOTO 5110
5080 PRINT ST(T,1);
5090 INPUT ": ";A$
5100 IF A$="N" OR A$="n" THEN ST(T,0)=0
5110 NEXT T
5120 PRINT
5130 PRINT "Nieuwe uitgangen:"
5140 PRINT
5150 S$=""
5160 INPUT ": ";S$
5170 IF S$="" THEN GOTO 5260
5180 GOSUB 4040:IF F=1 THEN 5150
5190 ZOEK=VAL(S$)
5200 GOSUB 4120
5210 IF G=1 THEN GOTO 5240
5220 PRINT "Element ";ZOEK;" is onbekend."
5230 GOTO 5150
5240 ST(PS,0)=1
5250 GOTO 5150
5260 RETURN
5270 REM ---Uitvoer---
5280 CLS:LOCATE 13
5290 PRINT "----Resultaten----"
5300 LOCATE 7,3
5310 PRINT "Ingangen:";
5320 LOCATE 26
5330 PRINT "Uitgangen:"
5340 PRINT
5350 G1=1:G2=1
5360 FOR T=1 TO TEL
5370 IF ST(T,2)=8 THEN LOCATE 6,G1+4:PRINT ST(T,1);": ";ST(T,
3):G1=G1+1
5380 IF ST(T,0)=1 THEN LOCATE 25,G2+4:PRINT ST(T,1);": ";ST(T
,3):G2=G2+1
5390 NEXT T
5400 RETURN
5410 REM ---Commando's---
5420 CLS
5430 LOCATE 15
5440 PRINT "----Menu----"
5450 PRINT:PRINT
5460 PRINT "S : Start simulatie"
5470 PRINT "WT: Opstellen waarheidstabel"
5480 PRINT "L : Laten zien van model"

```

```

5490 PRINT "C : Laten zien van schijfcatalogus"
5500 PRINT "NM: Nieuw model invoeren"
5510 PRINT "VM: Veranderen model"
5520 PRINT "VI: Verandeen ingangen en": PRINT TAB(5);"start s
imulatie"
5530 PRINT "DU: Definiëren uitgangen"
5540 PRINT "PU: Uitvoer laten zien"
5550 PRINT "OD: Opslaan van model op diskette"
5560 PRINT "OC: Opslaan van model op cassette"
5570 PRINT "H : Deze commandolijst laten zien"
5580 PRINT "ST: Verlaten van het programma"
5590 PRINT
5600 LOCATE ,23
5610 INPUT"# ";A$
5620 IF A$="S" THEN GOSUB 2770: GOTO 5780
5630 IF A$="WT" THEN GOSUB 2940: GOTO 5780
5640 IF A$="L" THEN GOSUB 4200: GOTO 5780
5650 IF A$="C" THEN FILES: PRINT: PRINT: GOTO 5780
5660 IF A$="NM" THEN GOTO 270
5670 IF A$="VM" THEN GOTO 1490
5680 IF A$="PU" THEN GOSUB 5270:GOTO 5780
5690 IF A$="VI" THEN GOSUB 4800:GOSUB 2770:GOTO 5780
5700 IF A$="DU" THEN GOSUB 4980:GOTO 5780
5710 IF A$="OD" THEN GOSUB 1160:GOTO 5780
5720 IF A$="OC" THEN GOSUB 1000 : GOTO 5780
5730 IF A$="H" THEN GOTO 5410
5740 IF A$="ST" THEN END
5750 LOCATE ,24
5760 PRINT "Niet toegestaan commando"
5770 PRINT
5780 GOTO 5600
5790 END

```

#### Checksums:

Gebruik het programma van paragraaf 4.6.

10:DB53	20:BAD4	30:2533	40:C2D4	50:1904	
60:36EC	70:CED4	80:E953	90:E718	100:E280	
110:D278	120:3A5B	130:0D7E	140:0CD7	150:A284	
160:B093				blok-checksum :	749B
170:5933	180:ABD4	190:030E	200:A370	210:34E4	
220:889E	230:3564	240:AEC2	250:B636	260:35D9	
270:0D27	280:B278	290:5BED	300:4D37	310:3510	
320:AA78				blok-checksum :	EFE7
330:7EAD	340:AE7A	350:86B5	360:B27A	370:C258	
380:B67A	390:9861	400:5512	410:6A0F	420:3043	
430:5063	440:9C5E	450:7E34	460:DABA	470:B7BB	
480:0D74				blok-checksum :	AF87
490:5EDF	500:FD34	510:B2E8	520:658A	530:6932	
540:FF85	550:E26B	560:01C5	570:6404	580:C0B5	
590:C6EF	600:CA98	610:DA77	620:AD41	630:9585	
640:AF01				blok-checksum :	3A63
650:C850	660:F5A1	670:F821	680:AB56	690:389A	
700:36D9	710:3677	720:6F76	730:6EDF	740:0D97	
750:8134	760:9944	770:38E4	780:E294	790:DBBD	
800:02E9				blok-checksum :	0835

810:0D47	820:3824	830:6DCA	840:5F86	850:38E4	
860:9DFD	870:06F2	880:9615	890:AF41	900:DE36	
910:5A04	920:05A2	930:3822	940:AB58	950:391A	
960:3959				blok-checksum :	7519
970:38F7	980:CDD5	990:74F6	1000:932D	1010:18C7	
1020:932A	1030:5545	1040:5510	1050:0E80	1060:AD9F	
1070:57E0	1080:11F8	1090:CDA7	1100:5838	1110:440B	
1120:655A				blok-checksum :	F998
1130:6599	1140:6537	1150:CCF6	1160:A32E	1170:1947	
1180:6724	1190:9C2B	1200:7547	1210:591F	1220:C94A	
1230:8275	1240:B4BD	1250:C708	1260:5840	1270:1220	
1280:E5A7				blok-checksum :	F85A
1290:B839	1300:440E	1310:661A	1320:6659	1330:65F7	
1340:6724	1350:CEF6	1360:04DE	1370:5387	1380:71E8	
1390:19A7	1400:18F6	1410:347B	1420:89F2	1430:3873	
1440:D502				blok-checksum :	0232
1450:67E4	1460:2436	1470:5A64	1480:6842	1490:D09A	
1500:7503	1510:A56E	1520:9394	1530:ADC9	1540:663E	
1550:7EA9	1560:047E	1570:95A3	1580:DA1C	1590:B3C9	
1600:2F19				blok-checksum :	7797
1610:6799	1620:64FD	1630:80FE	1640:972B	1650:582F	
1660:3070	1670:1606	1680:31FB	1690:9A17	1700:198F	
1710:661A	1720:2F37	1730:0480	1740:95D1	1750:3D19	
1760:6959				blok-checksum :	13EC
1770:8467	1780:1F22	1790:DF06	1800:CE53	1810:06D7	
1820:B5E5	1830:C25F	1840:B563	1850:CCDC	1860:12AF	
1870:49A5	1880:9C0D	1890:B9C9	1900:66EE	1910:19DF	
1920:F082				blok-checksum :	FC8A
1930:0488	1940:D175	1950:B7C9	1960:4449	1970:539F	
1980:FE26	1990:3C29	2000:0722	2010:9ADD	2020:64A3	
2030:046C	2040:A8C9	2050:64AE	2060:66D9	2070:4E1B	
2080:5ABE				blok-checksum :	CD05
2090:BFE5	2100:2DA7	2110:70C4	2120:5925	2130:33A7	
2140:FBF1	2150:4C25	2160:39A7	2170:3E25	2180:3DAD	
2190:D767	2200:EF21	2210:6912	2220:35AD	2230:804E	
2240:6793				blok-checksum :	E675
2250:2295	2260:3098	2270:80FE	2280:3EBC	2290:71C0	
2300:66E3	2310:A9C5	2320:CF76	2330:DF34	2340:6812	
2350:2299	2360:309A	2370:88FE	2380:39CC	2390:73C9	
2400:6763				blok-checksum :	B886
2410:E9C5	2420:D076	2430:04C3	2440:D0C1	2450:273E	
2460:D276	2470:826F	2480:A96E	2490:2B3E	2500:D076	
2510:05E3	2520:4BC4	2530:7A15	2540:3ECC	2550:682F	
2560:4BC7				blok-checksum :	995E
2570:7C15	2580:3EEC	2590:6937	2600:6892	2610:9736	
2620:89C6	2630:D2F6	2640:4750	2650:ABCA	2660:2C3E	
2670:D4F6	2680:21E0	2690:8BC6	2700:8CFE	2710:3ED4	
2720:A1BB				blok-checksum :	C3B2
2730:D3F6	2740:C554	2750:A1EB	2760:D576	2770:3004	
2780:1A37	2790:1A72	2800:1EDE	2810:9698	2820:1412	
2830:08F3	2840:3964	2850:2E96	2860:8774	2870:D693	
2880:6A4B				blok-checksum :	EE78
2890:75CF	2900:1857	2910:1EFA	2920:9A28	2930:D5F6	
2940:3E4C	2950:A942	2960:6BF5	2970:6BE4	2980:7E97	
2990:E2E4	3000:2ADA	3010:CD80	3020:9817	3030:6719	
3040:32D6				blok-checksum :	95D8

3050:E157	3060:6C1F	3070:3AA5	3080:6864	3090:4473	
3100:4401	3110:FADF	3120:0361	3130:7936	3140:218D	
3150:596D	3160:55F4	3170:68E4	3180:6037	3190:1DA4	
3200:9BD9				blok-checksum :	CDB5
3210:0FF5	3220:9C17	3230:6819	3240:3433	3250:199F	
3260:3DE0	3270:C55C	3280:6964	3290:9910	3300:A170	
3310:A029	3320:AC94	3330:02E6	3340:68D9	3350:5971	
3360:5158				blok-checksum :	1FCA
3370:2D07	3380:69E4	3390:6A64	3400:50C3	3410:68A2	
3420:6924	3430:3492	3440:D376	3450:BAFF	3460:9840	
3470:A45F	3480:3456	3490:A158	3500:6E1F	3510:5A7B	
3520:6904				blok-checksum :	7333
3530:5C7A	3540:5308	3550:D39A	3560:3505	3570:B6DE	
3580:53CB	3590:4F56	3600:04D6	3610:B308	3620:D677	
3630:13F7	3640:9C25	3650:4BC6	3660:6AE3	3670:57B9	
3680:BF16				blok-checksum :	9672
3690:519D	3700:4DB9	3710:D475	3720:D41F	3730:3744	
3740:1A73	3750:C158	3760:CBFD	3770:6BA4	3780:C4AC	
3790:9176	3800:A92A	3810:4F04	3820:55E4	3830:13B9	
3840:6B5A				blok-checksum :	E537
3850:597D	3860:F158	3870:3207	3880:6C64	3890:6CE4	
3900:6AD9	3910:6B64	3920:D776	3930:6E61	3940:76E9	
3950:D42A	3960:6CA4	3970:63D7	3980:DA76	3990:E29F	
4000:BE16				blok-checksum :	DAB6
4010:319D	4020:1963	4030:D0F6	4040:BE24	4050:1992	
4060:7B03	4070:D2F6	4080:8E68	4090:0B89	4100:7EB2	
4110:10C7	4120:2C56	4130:1996	4140:1BDF	4150:88F2	
4160:26B9				blok-checksum :	43BF
4170:28CA	4180:D413	4190:D4F6	4200:6BEC	4210:19C7	
4220:A32A	4230:1AE9	4240:69A4	4250:ECAE	4260:6A24	
4270:B7F7	4280:6AA4	4290:5476	4300:1BE1	4310:4158	
4320:9510				blok-checksum :	F6AC
4330:97C4	4340:69E4	4350:6A64	4360:D576	4370:745E	
4380:0212	4390:AC25	4400:EFDA	4410:A62A	4420:6010	
4430:5701	4440:5C6C	4450:8CCF	4460:7763	4470:AD50	
4480:196D				blok-checksum :	62EB
4490:E7C2	4500:6EE2	4510:507D	4520:EFE3	4530:5481	
4540:6C31	4550:AFEC	4560:7E31	4570:0ACF	4580:3601	
4590:6C22	4600:6AA4	4610:D5F6	4620:AF40	4630:04FF	
4640:2790				blok-checksum :	D5DB
4650:3F22	4660:9F15	4670:013C	4680:A226	4690:6CA3	
4700:6AE4	4710:B649	4720:D776	4730:8733	4740:1A3D	
4750:013C	4760:941F	4770:6CA3	4780:6CE4	4790:DAF6	
4800:5D3C				blok-checksum :	B644
4810:1A87	4820:BD37	4830:5217	4840:6CA4	4850:2159	
4860:667A	4870:08BF	4880:EBD1	4890:D8FA	4900:F4D1	
4910:B79A	4920:084C	4930:BFFD	4940:9ED9	4950:6D24	
4960:6DA4				blok-checksum :	5C8E
4970:DBF6	4980:F51A	4990:1B27	5000:A12A	5010:7169	
5020:4673	5030:B112	5040:5BA8	5050:69E4	5060:7158	
5070:809C	5080:EFDB	5090:AA1C	5100:8279	5110:6924	
5120:69A4				blok-checksum :	BCF8
5130:1FF4	5140:6A24	5150:686A	5160:9A40	5170:9C3B	
5180:BAD5	5190:0513	5200:94FE	5210:5849	5220:9318	
5230:4FB9	5240:51FC	5250:53B9	5260:D676	5270:52D3	
5280:72AE				blok-checksum :	7CCF

5290:1BD4	5300:9D35	5310:21B1	5320:5281	5330:5618	
5340:6B24	5350:9831	5360:D158	5370:DE4F	5380:2280	
5390:6C24	5400:D576	5410:76C4	5420:1A57	5430:587E	
5440:E924				blok-checksum :	62C3
5450:09FF	5460:1DF6	5470:B05A	5480:CF9B	5490:A324	
5500:E1C9	5510:EE1F	5520:13ED	5530:398F	5540:5332	
5550:AA06	5560:9E16	5570:B549	5580:1A24	5590:6D64	
5600:B515				blok-checksum :	FBBB
5610:E8AC	5620:9234	5630:A28B	5640:6AB5	5650:A27A	
5660:46BD	5670:8DEA	5680:A010	5690:8E67	5700:C00A	
5710:4A8B	5720:2A0C	5730:3797	5740:0514	5750:D116	
5760:7AE9				blok-checksum :	6E88
5770:6DE4	5780:6DC3	5790:1B24		blok-checksum :	D831

## 6 SPELLETJES

Alhoewel dit boek allereerst is bedoeld om serieuze toepassingen voor de MSX te geven willen wij u het genot van enige spelletjes niet onthouden.

Tevens krijgt u hiermede inzicht in een aantal typische mogelijkheden van de MSX.

Persoonlijk hebben we het meeste plezier beleefd aan het spel EXPLOSIË ....we kennen zelfs enkele personen die er volledig aan verslaafd zijn geraakt....

### 6.1 HET LETTERSPEL

Dit spelletje munt uit door zijn eenvoud en dat nu is doorgaans het kenmerk van uitdagende spelletjes.

De computer geeft ons steeds de letters A t/m J en wel in een willekeurige volgorde. Iedere letter heeft daarbij een unieke positie; de eerste letter komt met 1 overeen, de tweede met 2, etc. We kunnen de volgorde van de letters nu op een zeer speciale manier wijzigen. We geven namelijk steeds een letter door middel van zijn positienummer op en de letter die voor de aangegeven letter staat zal nu vooraan worden geplaatst en de letter direkt na de zo aangegeven letter komt nu achteraan te staan.

Het voorbeeld verduidelijkt dit.

*Voorbeeld:*

\*\*\*\* LETTERSPEL \*\*\*\*

IDCFBGJHAE

Zet : 5

De eerste keer wordt een 5 ingevoerd. De letter B wordt vooraan gezet en de hieropvolgende letter achteraan.

\*\*\*\* LETTERSPEL \*\*\*\*

BIDCFJHAEG

Zet : 4

De tweede keer wordt een 4 ingevoerd, wat tot gevolg heeft dat de C naar voren en de F naar achteren verhuist.

\*\*\*\* LETTERSPEL \*\*\*\*

CBIDJHAEGF

Zet :

Probeer de juiste (alfabetische) volgorde te vinden in een zo gering mogelijk aantal beurten.

*Programma:*

```
10 ' *** LETTERSPEL ***
20 DEFINT A-Z:B$="LETTERSPEL"
30 CLS:A$="ABCDEFGHJIJ"
40 T=0:C=RND(-TIME)
50 ON INTERVAL=300 GOSUB 310
60 INTERVAL ON
70 FOR C=1 TO 10
80 Z=INT(RND(1)*9+1)
90 GOSUB 280
100 NEXT C
110 LOCATE 6,2
120 PRINT"**** "B$" **** "
130 LOCATE 4,6
140 PRINT A$
150 LOCATE 6,8
160 PRINT"Zet :";
170 C$=INKEY$:IF C$="" THEN 170
180 Z=VAL(C$):PRINT Z
190 IF Z=0 THEN 150
200 T=T+1
210 GOSUB 280
220 IF A$="ABCDEFGHJIJ"THEN PRINT"GOED IN" T "BEURTE
N":GOTO 240
230 GOTO 130
240 PRINT:PRINT"Nog eens (j/n)"
250 B$=INKEY$:IF B$="" THEN 250
260 IF B$="j" OR B$="J" THEN 10
```



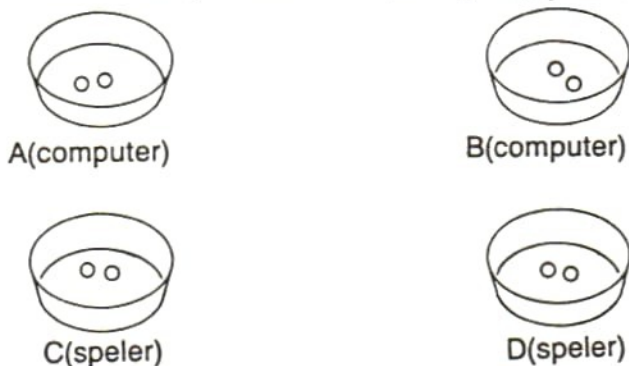
```

270 END
280 ' ** manipulatie aan A$ **
290 A$=MID$(A$,Z,1)+LEFT$(A$,Z-1)+RIGHT$(A$,9-Z)+
MID$(A$,Z+1,1)
300 RETURN
310 B$=RIGHT$(B$,1)+LEFT$(B$,9)
320 P=POS(1):Q=CSRLIN
330 LOCATE 11,2
340 PRINT B$:LOCATE P,Q
350 RETURN

```

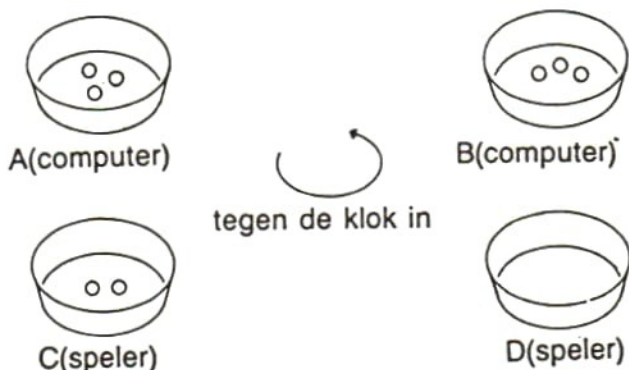
## 6.2 MINI MANCALA

Mini mancala is een originele creatie van C. Freerlink. Het is gebaseerd op een oud Arabisch spel, waarbij stenen in kommetjes worden verdeeld. Er zijn vier bakjes: A en B zijn van de computer en C en D zijn van U. Als het spel begint liggen er twee stenen in elk bakje.



Om de beurt nemen de spelers de stenen uit één van hun eigen bakjes en verdelen ze één voor één tegen de klok in over de andere drie bakjes.

U kunt er bijvoorbeeld voor kiezen om de stenen uit bakje D te verdelen:



Na deze zet is bakje D leeg. Er zal trouwens altijd minstens één bakje leeg zijn want gedurende iedere zet mogen de te verdelen stenen nooit in het bakje waaruit ze gekomen zijn worden teruggelegd.

Om te winnen moeten alle stenen in een van Uw eigen bakjes liggen. Zo kan de tegenstander immers geen enkele zet meer doen.

We kunnen allereerst een moeilijkheidsgraad kiezen (1, 2 of 3: 1 is het gemakkelijkst). Als u met moeilijkheidsgraad 1 begint en u wilt zal de volgende ronde automatisch in moeilijkheidsgraad 2 gespeeld worden. Bovendien mogen we aangeven wie mag beginnen. Als u niet snel genoeg reageert komt het programma in de demo-mode. Door op de aangegeven toets te drukken kunt u met spelen beginnen.

Als U aan de beurt bent vraagt de computer van welk bakje U de stenen wilt verspreiden. Uiteraard geeft de computer steeds aan wat zijn eigen zet is. Na iedere zet toont de computer hoe de stenen over de bakjes verdeeld zijn.

Het is in feite verbazingwekkend dat een dergelijk ingewikkeld spel met zo'n kort programma beschreven kan worden.

Tenslotte vestigen we de aandacht op het diagram dat hier is afgebeeld. Dit diagram toont alle mogelijke zetten, kortom alle strategische wegen die we in dit zo boeiende spel kunnen bewandelen. Het is een fraai voorbeeld van een met name strategisch spel.

#### *Programma:*

```
10 SCREEN0:WIDTH40
20 REM ----mini mancala ----
30 A=RND(-TIME):W1=1500:W2=500
40 GOTO590
50 REM initialisatie
60 DEF FNSP(X)=15+80*(XAND2)+6*(XAND1)
70 V=-1:FOR C=1 TO 2:FOR X=0 TO 3:GOSUB1060:PLAY
"l16t255m400s1o6c":NEXT X:NEXT C
80 RESTORE 110:FOR I=0TO3
90 READSU(I)
100 NEXTI
110 DATA2,0,3,1
120 RETURN
130 REM schermlayout
140 SCREEN 2:COLOR 1
150 OPEN "GRP:" FOR OUTPUT AS #1
160 IF AU THEN COLOR1:PRESET(48,5):PRINT#1,"D E M
O Stop=<F1>":ON KEY GOSUB 1150:KEY(1) ON
170 N=0:FOR X=90 TO 165 STEP 75
180 FOR Y=40 TO 100 STEP 60
190 PLAY"s9m60300l1o1c"
200 CIRCLE(X,Y+6),36,14,,.8
210 PAINT(X,Y),14,14
220 CIRCLE(X,Y),30,1,,.69
```

```

230 PAINT(X,Y),1,1
240 N=N+1:PRESET(X-3,Y+25),14:PRINT #1,MID$("ACBD",N,1)
250 NEXT Y:NEXT X
260 PSET(8,145):DRAW("c2s32r29d2129u2"):PAINT(20,150),2,2
270 RETURN
280 REM invoeren
290 IF B(2)=0 THEN PRESET(15,150):PRINT#1,"Geen keus. Je gooit bak D":FOR WL=1 TO W1/3:NEXT:S=3:RETURN
300 IF B(3)=0 THEN PRESET(15,150):PRINT#1,"Geen keus. Je gooit bak C":FOR WL=1 TO W1/3:NEXT:S=2:RETURN
310 PRESET (15,150):PRINT #1,"Jouw beurt, kies bak C of D":IF AU THEN S=2-(RND(1)<.5):FOR WL=1 TO 500:NEXT WL:RETURN
320 IF INKEY$<>"" THEN 320
330 A$=INKEY$:IF A$="" THEN 330
340 S=INSTR(1," cCdD",A$):IF S=0 THEN 330
350 S=S\2+1:IFS=1THEN 330
360 RETURN
370 REM beurt computer
380 IF B(0)=0 THEN PRESET(15,150):PRINT#1,"Ik kan alleen bak B spelen":S=1:FOR WL=1 TO W1:NEXT:RETURN
390 IF B(1)=0 THEN PRESET(15,150):PRINT#1,"Ik kan alleen bak A spelen":S=0:FOR WL=1 TO W1/3:NEXT:RETURN
400 PSET(15,150):PRINT #1,"Mijn beurt. ";
410 FOR WL=1 TO W1/3:NEXT
420 IF LE<3 AND RND(TIME)*LE<.6 THEN S=INT(RND(TIME)*2):GOTO 450
430 Q=B(2)+10*(B(1)+10*B(0))
440 S=1
450 IF Q=143 OR Q=134 OR Q=611 OR Q=116 THEN S=0
460 IF B(S)=0 THEN S=1-S
470 PRINT #1,"Ik kies bak "+MID$("AB",S+1,1)
480 RETURN
490 REM verdelen tegen de klok in
500 D=S
510 FOR C=1 TO B(S)
520 FOR WL=1 TO W2:NEXT:PLAY "m2600s3o4t120"+MID$("cdefgab",B(S),1)
530 D=SU(D):IF D=S THEN 530
540 X=S:V=0:GOSUB1060

```

```

550 X=D:V=-1:GOSUB1060
560 NEXT C
570 COLOR2:PSET(15,150):PRINT #1,STRING$(27,219):
PSET(15,150):COLOR 1
580 RETURN
590 REM hoofdprogramma
600 GOSUB 900:REM spelnivo
610 GOSUB130:REM scherm layout
620 GOSUB 50:REM initialisatie
630 IF PQ<3 THEN 690
640 GOSUB280:REM zet invoeren
650 GOSUB490:REM verdelen
660 IF B(3)=8 THEN 730:REM winst
670 W2=W2-25:IF W2<60 THEN W2=60
680 W1=W1-25:IF W1<300 THEN W1=300
690 GOSUB370:REM beurt computer
700 GOSUB490:REM verdelen
710 IF B(0)=8 THEN 780:REM verliest
720 GOTO640
730 REM speler winst
740 PLAY "o3t100cc.ccc."
750 GOSUB 570:PRINT#1,"Je hebt gewonnen !!!":FOR
WL=1 TO W1:NEXT:GOSUB570
760 LE=LE+1+(LE=3):PRINT#1,"Nivo";LE;
770 GOTO 820
780 REM speler verliest
790 PLAY "t255o5co4bagfedc"
800 GOSUB570:PRINT#1,"Ik heb van je gewonnen !!!"
:FORWL=1 TO W1:NEXT:GOSUB570
810 LE=LE-1-(LE=1):PRINT#1,"Nivo";LE;
820 PRINT#1,". Begin jij? (j/n)"
830 IF AU THEN PQ=3+(RND(1)>.5):FOR WL=1TO300:NEX
T WL:GOTO860
840 A$=INKEY$:IF A$="" THEN 840
850 PQ=INSTR(1,"nNjJ",A$):IF PQ=0 THEN 840
860 V=0:FOR X=0 TO 3
870 IF B(X)=0 THEN 880 ELSE GOSUB1060:GOTO870
880 NEXT X
890 GOSUB570:GOTO 620
900 REM spelnivo
910 KEY OFF:SCREEN0,0,0,1:COLOR1,12,12
920 WL=100:LOCATE5,4:A$="M I N I - M A N C A L A"
:GOSUB1050:LOCATE3,12
930 WL=10:A$="Kies het spelnivo (1--3) ":GOSUB105
0
940 T=0

```

```

950 A$=INKEY$:IF A$="" THEN T=T+1
960 IF T=400 THEN AU=1: A$="2"
970 IF A$="" THEN 950
980 IF INSTR(1,"123",A$)=0 THEN 950
990 LE=VAL(A$):PLAY "s7m1500o3b":PRINTMID$("laag
middelhoog",1+6*(LE-1),6):LOCATE3,14
1000 COLOR,2,2:A$="Begint U ? (j/n)":GOSUB1050
1010 IF AU THEN A$="n":FOR WL=1 TO 900:NEXT WL:GO
TO1030
1020 A$=INKEY$:IF A$="" THEN 1020
1030 PQ=INSTR(1,"nNjJ",A$):IF PQ=0 THEN 1020
1040 PLAY "s3m4000o1b":CLS:COLOR15,6,6:RETURN
1050 FOR X=1 TO LEN(A$):PRINTMID$(A$,X,1);:FORW=1
TOWL:NEXT W:NEXTX:RETURN
1060 REM bijwerken bak
1070 B(X)=B(X)-(V=-1)+(V=0)
1080 P=B(X)-(V=0)
1090 X1=78+75*(X MOD 2)+5*VAL(MID$("12031435",P,1
))
1100 Y1=31-60*(X>1)+10*VAL(MID$("01102120",P,1))
1110 CL=1-5*V
1120 CIRCLE(X1,Y1),4,CL
1130 PAINT(X1,Y1),CL,CL
1140 RETURN
1150 RUN

```

### 6.3 ABBA-SPEL

Omdat de naam van de bekende popgroep uit alleen maar A's en B's bestaat, is deze naam zeer geschikt om er een puzzel van te maken:

Als we het programma starten zien we een vierkant van 4 bij 4 met enkel A's en B's. Met behulp van de cursortoetsen kunnen we naar een bepaalde letter in dit vierkant gaan. Door op de RETURN-toets te drukken kunnen we een bepaalde letter van het vierkant aangegeven. Door zo'n letter kunnen we steeds een horizontale en verticale lijn trekken. Alle letters die op deze lijnen liggen veranderen: een A wordt een B en een B wordt een A.

Het voorbeeld verduidelijkt de situatie.

Het is de bedoeling om uiteindelijk de volgende situatie te bereiken:

```

ABBA
ABBA
ABBA
ABBA

```

Veel succes!

Programma:

```
10 SCREEN0:WIDTH40
20 KEYOFF:COLOR15,1,1
30 A=RND(-TIME):CLS:LOCATE8,15:PRINT"Een ogenblik
je a.u.b. ";
40 M=1
50 DEF FNP(X,Y)=X+(Y-1)*4
60 DIMZ(16,7),B(16),TR(16),TS(16),RP(25),PS(16)
70 FOR X=1 TO 4:FOR Y=1 TO4
80 N=FNP(X,Y):L=1
90 FOR R=1 TO 4:Z(N,L)=FNP(R,Y):L=L+1:NEXT R
100 FOR K=1 TO 4:IF Y=K THEN 120
110 Z(N,L)=FNP(X,K):L=L+1
120 NEXTK
130 NEXTY:NEXTX
140 AB=0:FOR K=1 TO 4:T=1+2*((K=1)OR(K=4))
150 FORR=0TO12STEP4
160 C=K+R:B(C)=T:TR(C)=T
170 NEXTR:NEXTK
180 COLOR15,1,1:CLS
190 IF NOT QQ THEN QQ=1=1:SCREEN1,2:GOSUB 530
200 FOR C=1 TO M:P=INT(RND(1)*16):P=P+(P=17):RP(C)
)=P:GOSUB510:NEXT C
210 FOR X=1 TO16:TS(X)=B(X):PS(X)=4+3*(B(X)=-1):N
EXT X
220 XO=90:YO=15:SP=0:FOR Y=1 TO 4:FOR X=1 TO 4:H=
XO+(X-1)*20:V=YO+(Y-1)*23:N=FNP(X,Y)
230 SP=SP+1:PUT SPRITE SP,(H,V),15,4+3*(B(N)=-1):
NEXTX:NEXTY
240 P=1
250 N=4+3*(B(P)=-1)
260 PUT SPRITE P,,4,N
270 A$=INKEY$:IF A$="" THEN 270
280 PUT SPRITE P,,15,N
290 A=ASC(A$)
300 IF A=28 THEN P=P+1
310 IF A=31 THEN P=P+4
320 IF P>16 THEN P=1
330 IF A=29 THEN P=P-1
340 IF A=30 THEN P=P-4
350 IF P<1 THEN P=16
360 IF A<>13 THEN 250
370 FORT=1TO3:FORTT=1TO7:SP=Z(P,TT)
380 PS(SP)=PS(SP)-B(SP)
```

```

390 PUT SPRITE SP,, ,PS(SP)
400 NEXT TT:NEXT T
410 GOSUB510
420 A=0:FOR T=1 TO16:A=A OR (B(T)<>TR(T)):NEXT T
430 AB=AB+1:LOCATE 2,17:PRINT"Beurtnummer:";AB+1
440 IF A THEN 250
450 J$="":IF AB>1 THEN J$="en"
460 CLS:LOCATE 1,15:PRINT"Het is je in";AB;"beurt
"+J$:PRINT" gelukt.":PRINT:M=M+1:PRINT" Probeer n
u";M;"verschuivingen"
470 PRINT:PRINT" Sla een toets ,aan"
480 IF INKEY$="" THEN480
490 GOTO140
500 END
510 REM hussel
520 FORT=1TO7:B(Z(P,T))=-B(Z(P,T)):NEXTT:RETURN
530 SP=1:REM sprite maken
540 READA:IF A=0 THEN RETURN
550 V$="":FORX=1 TOA:READV:V$=V$+CHR$(V):NEXTX:SP
RITE$(SP)=V$:SP=SP+1:GOTO540
560 END
570 DATA 32 :REM Abba 1
580 DATA &H0,&H3,&H4,&H4,&H8,&H8,&H8,&H1F,&H10,&
H10,&H10,&H10,&H10,&H10,&H38,&H0,&H0,&HC0,&H20,&H
20,&H10,&H10,&H10,&HF8,&H8,&H8,&H8,&H8,&H8,&H8,&H
1C,&H0
590 DATA 32 :REM Abba 2
600 DATA &H0,&H3,&H4,&H4,&H8,&H8,&HC,&H13,&H10,&
H10,&H10,&H10,&H10,&H10,&H30,&H0,&H0,&HC0,&H20,&H
20,&H20,&H10,&H30,&HD8,&H8,&H8,&H8,&H8,&H8,&H8,&H
30,&H0
610 DATA 32 :REM Abba 3
620 DATA &H0,&HF,&H4,&H4,&H8,&H8,&HE,&H11,&H10,&
H10,&H10,&H10,&H10,&H10,&H31,&H0,&H0,&HC0,&H20,&H
20,&H20,&H20,&H60,&H90,&H8,&H8,&H8,&H8,&H8,&H10,&
HE0,&H0
630 DATA 32 :REM Abba 4
640 DATA &H0,&H3F,&H10,&H10,&H10,&H10,&H1F,&H10,&
&H10,&H10,&H10,&H10,&H10,&H3F,&H0,&H0,&HC0,&
H20,&H20,&H20,&H20,&HC0,&H20,&H10,&H8,&H8,&H8,&H8
,&H10,&HE0,&H0
650 DATA 0

```

## 6.4 CODE KRAKEN

Het zou hier om een spannend spionageverhaal kunnen gaan waarbij we zo snel mogelijk een brandkast moeten kraken om aan belangrijke informatie te komen. De code van de brandkast kent U natuurlijk nog niet en het is de vraag in hoeveel beurten U die kunt vinden. Doe er niet te lang over ... stel dat er iemand komt!

De code is opgeslagen in een computer en we weten dat die code bestaat uit drie series van 20 enen en nullen, waarbij we steeds uitgaan van de volgende serie:

```
10100101011101010110
```

Die drie series zijn door „cyclische verwisseling” verkregen, in gewoon Nederlands wil dat zeggen dat getallen van het linkerruimte naar rechts worden verplaatst. Hier zien we een voorbeeld van 3 van dergelijke series tesamen met de som van die getallen voor iedere kolom.

```
001010111101010110101
01110101011010100101
01001010111010101101
-----
02212122223030311303
```

Deze series zijn natuurlijk voor U verborgen. Sterker nog U moet deze series weer zien terug te vinden. Als het spel begint krijgt U ook drie rijen getallen te zien, deze zijn ook verkregen door cyclische verwisseling van de reeks ... helaas deze verwisseling is willekeurig. Een voorbeeld: als de bovenste rij 2 posities is opgeschoven, de middelste 4 en de onderste 8, dan zien we op het beeldscherm:

```
10101110101011010100
01010110101001010111
11101010110101001010
-----
20000112111123310122
```

Merk op dat we nu niet de som van iedere nieuwe kolom te zien krijgen, maar juist het verschil tussen deze som en de eerste (verborgen) som.

Voor de eerste kolom geldt bijvoorbeeld dat de verborgen som 0 is en aangezien de som op het beeldscherm 2 is krijgen we nu  $2 - 0 = 2$  te zien. Bij de tweede kolom zijn zowel de verborgen som als de nieuwe som 2 dus we zien  $2 - 2 = 0$ . We moeten nu steeds de rijen verschuiven en wel tot ze precies gelijk zijn aan de verborgen series. In dit geval zijn alle verschillen natuurlijk 0, kortom de onderste serie op het beeldscherm toont dan alleen maar nullen.

Door de cursor op een bepaalde positie in een bepaalde rij te zetten en vervolgens op RETURN te drukken komt het getal dat op de aangegeven positie stond vooraan de rij te staan. De hele rij is dus opgeschoven. De cursor staat nu op een zodanige positie dat als we opnieuw RETURN geven zonder iets aan de cursorpositie te veranderen, de oude situatie, dus die van voor de laatste zet, teruggekregen wordt.

Hoeveel beurten heeft U nodig om de code te kraken? We kennen super-spionnen die de klus in 10 beurten klaren



Programma:

```
10 KEYOFF:SCREEN 0:WIDTH 40:COLOR15,1,1
20 CLEAR 2000:A=RND(-TIME)
30 LOCATE 8,13:PRINT"Een ogenblikje a.u.b."
40 ON KEY GOSUB 590:KEY(1) ON
50 DEFFNM(A)=A+20*(A>20)-20*(A<1)
60 DEFFNV$(P,I)=MID$("10100101011101010110",FNM(A
(I)-1+P),1)
70 DEFFNV(P,I)=VAL(FNV$(P,I))
80 DIM S$(20):FOR I=1 TO 20:A(0)=I:H$="":FOR P=1
TO 20:H$=H$+FNV$(P,0):NEXT P:S$(I)=H$:NEXT I
90 SCREEN 1,2:WIDTH 32:COLOR15,1,1
100 FOR I=1 TO 3:A(I)=1:NEXT I:IF INKEY$=" " THEN
120
110 FOR I=1 TO 3:A(I)=INT(RND(1)*20+1):A(I)=A(I)+
(A(I)=21):NEXT I
120 SOM$="":FOR P=1 TO 20:S=0:FOR I=1TO3:S=S+FNV(P
,I):NEXT I:SOM$=SOM$+RIGHT$(STR$(S),1):NEXT P:S$(
0)=SOM$:A(5)=0
130 FOR I=1 TO 3:A(I)=INT(RND(1)*20+1):A(I)=A(I)+
(A(I)=21):GOSUB270:NEXT I:LOCATE 4,12:PRINTSTRING
$(20,192)
140 '
150 GOSUB310:'somverschil
160 GOSUB430:'maak cursor
170 B$=INKEY$:IF B$="" THEN 170 ELSE B=ASC(B$)
180 IF B=13 THEN GOSUB 460
190 IF B=28 THEN PS=FNM(PS+1)
200 IF B=29 THEN PS=FNM(PS-1)
210 IF B=30 THEN RS=RS-1-(RS=1)
220 IF B=31 THEN RS=RS+1+(RS=3)
230 IF Q THEN 530
240 GOSUB440:GOTO 170
250 '
260 END
270 REM print A$(i)
280 LOCATE 4,5+2*I,0:FORT1=1 TO50:NEXT T1
290 PRINT S$(A(I))
300 RETURN
310 REM print somverschil
320 LOCATE 4,15,0
330 Q=0:FOR P1=1 TO 20:S=0:FOR I1=1TO3:S=S+FNV(P1,
I1):NEXT I1:S=ABS(S-VAL(MID$(S$(0),P1,1))):Q=Q OR
(S<>0):PRINTRIGHT$(STR$(S),1);:NEXT P1
```

```

340 Q=NOT Q:RETURN
350 REM spritemaker
360 RESTORE 400 :S=1
370 V$="": READ A: IF A=0 THEN RETURN
380 FOR X=1 TO A:READ B:V$=V$+CHR$(B):NEXT X
390 SPRITE$(S)=V$:S=S+1:GOTO 370
400 DATA 32 :REM cursor
410 DATA &H7F,&HC1,&H80,&H80,&H80,&H80,&H80,&H80,
&H80,&H80,&H80,&HC1,&H7F,&H0,&H0,&H0,&H0,&H80,&H
80,&H80,&H80,&H80,&H80,&H80,&H80,&H80,&H80,&
H0,&H0,&H0,&H0
420 DATA 0
430 GOSUB350:X0=30:Y0=52:RS=1:PS=1
440 PUTSPRITE1,(X0+(PS-1)*8,Y0+(RS-1)*16),5,1
450 RETURN
460 REM schuiven
470 IF PS=1 THEN 520
480 I=RS:M=A(I):MN=FNM(PS-1+M):ST=SGN(PS-10)+(PS=
10):PS=22-PS
490 A(I)=FNM(A(I)-ST):GOSUB270
500 IF A(I)<>MN THEN 490
510 GOSUB310
520 AB=AB+1:RETURN
530 REM einde spel
540 SCREEN0:LOCATE3,8:PRINT"Het is je in";AB;"beu
rt";:IF AB>1 THENPRINT"en";
550 PRINT" gelukt.
560 LOCATE 3,15:PRINT"Nog een keer? (j/n)";
570 A$=INPUT$(1):IF A$="n" OR A$="N" THEN END
580 AB=0:GOTO 90
590 RUN

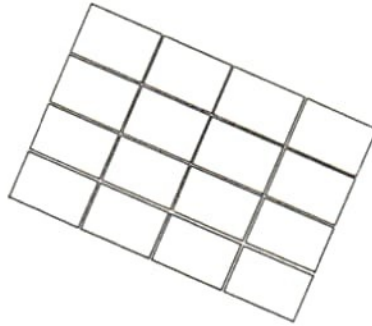
```

## 6.5 EXPLOSIE

Het spel EXPLOSIE is stellig een spel waaraan je verslaafd kunt raken, dus als het niet meer zo goed gaat in het zakenleven....

Het gaat om een bordspel waarbij men zelf mag kiezen of het bord  $3 \times 3$ , dan wel  $4 \times 4$  velden heeft. Wordt de keuze niet snel gemaakt dan raakt het programma in de demomode. Hier is uit te komen door op de aangegeven toets te drukken.

Laten we als voorbeeld eens uitgaan van een  $4 \times 4$  bord:



Het is belangrijk te bedenken dat ieder veld een bepaalde capaciteit heeft. Deze capaciteit is gelijk aan het aantal burens dat met een gehele zijde aangrenzend is. Zo heeft een veld in een hoekpunt een capaciteit van 2 omdat het precies twee burens heeft. Een veld aan de rand heeft drie burens en dus een capaciteit van 3 en een veld in het midden heeft een capaciteit van 4.

De spelregels zijn als volgt:

- om beurten zetten de computer en U een steen op een veld, en wel op een nog leeg veld of op een eigen veld (hier staan al eigen stenen op). We zetten een steen op een veld door met de cursortoetsen naar dat veld toe te gaan en vervolgens op RETURN te drukken. Het is wel even wennen welke toets voor welke richting gebruikt moet worden.
- als door het zetten van een steen op een veld het aantal stenen gelijk wordt aan de capaciteit van dat veld, dan ontploft dat veld. De inhoud wordt dan verspreid over de aanliggende burens die nu allen de kleur van de speler krijgen. Zo kan men velden op de tegenstander veroveren.
- de winnaar is die speler die alle vakjes bezit of een oneindige keten explosies veroorzaakt

*Programma:*

```
10 REM *** EXPLOESIE ***
20 DIM CX(4,4),CY(4,4),SB(5,5),ST(5,5),RB(5,5),SQ
(5,5)
30 A=RND(-TIME):KEY OFF:DEFINT A-Z
40 XO=115:YO=140:REM positie bord
50 RX=1:RY=1
60 COLOR 1,5,5
70 REM ---explosie---
80 SCREEN0
90 LOCATE10,8:PRINT"E X P L O S I E"
100 REM initialisatie
110 GOSUB 360
120 REM spelen
```

```

130 REM
140 IF AU THEN GOSUB980:GOTO 170
150 IF PL=1 THEN GOSUB590:NT=NT+1
160 IF PL=-1 THEN GOSUB980
170 PL=-PL
180 IF NOT ET THEN 140
190 REM eind
200 COLOR 4:PRESET(15,170):PRINT#1,"GEEXPLODEERD
!! Jij ";:IF PL=-1 THEN PRINT#1,"wint" ELSE PRINT
#1,"verliest"
210 PLAY"s14m10o111cc","s13m11o211dd"
220 IF PLAY(0) THEN 220
230 FOR WL=1 TO 600:NEXT WL:RUN
240 REM kop RB > SB
250 FOR X=1 TO SI
260 FOR Y=1 TO SI
270 SB(X,Y)=RB(X,Y)
280 NEXT Y:NEXT X
290 RETURN
300 REM kop SB > RB
310 FOR X=1 TO SI
320 FOR Y=1 TO SI
330 RB(X,Y)=SB(X,Y)
340 NEXT Y:NEXT X
350 RETURN
360 REM initialisatie
370 LOCATE 5,15
380 PRINT"Welke bordgrootte? (3 of 4)"
390 AU=1-0:T=0
400 A$=INKEY$:IF A$="" THEN T=T+1
410 IF T=300 THEN AU=1=1:PL=1:ON KEY GOSUB 1880:K
EY(1) ON:SI=4+(RND(1)>.5):GOTO450
420 IF A$="" THEN 400
430 IF A$<>"3" AND A$<>"4" THEN 400
440 SI=VAL(A$)
450 PLAY "o5s3m3000c"
460 FOR X=1 TO SI
470 FOR Y=1 TO SI
480 ST(X,Y)=4+(X=1)+(X=SI)+(Y=1)+(Y=SI)
490 NEXT Y
500 NEXT X
510 PRINT:PRINT:PRINT"           Wil jij beginnen? (j/n
)
520 IF AU THEN 550
530 A$=INKEY$:IF A$="" THEN 530
540 PL=INSTR(1,"jJnN",A$):IF PL=0 THEN 530

```

```

550 PLAY "o3s3m3000c"
560 IF PL<3 THEN PL=1 ELSE PL=-1
570 GOSUB 1390
580 RETURN
590 REM zet speler
600 GOSUB 1670
610 IF RB(MX,MY)<0 THEN 600
620 GOSUB 240
630 X=MX:Y=MY:DI=-1
640 GOSUB 670
650 GOSUB 300
660 RETURN
670 REM explosie
680 SB(X,Y)=SB(X,Y)+PL
690 NE=0
700 IF DI THEN X1=X:Y1=Y:GOSUB 930
710 XP=0
720 FOR Y=1 TO SI
730 FOR X=1 TO SI
740 IF ABS(SB(X,Y))<ST(X,Y) THEN 850
750 XP=-1
760 NE=NE+1
770 SB(X,Y)=SB(X,Y)-ST(X,Y)*PL
780 IF NOT DI THEN 810
790 X1=X:Y1=Y
800 GOSUB 930
810 X1=X:Y1=Y-1:GOSUB 890
820 X1=X+1:Y1=Y:GOSUB 890
830 X1=X:Y1=Y+1:GOSUB 890
840 X1=X-1:Y1=Y:GOSUB 890
850 NEXT X:NEXT Y
860 ET=(NE>1.5*SI*SI)
870 IF XP AND NOT ET THEN 710
880 RETURN
890 REM optellen bij buren
900 SB(X1,Y1)=PL*(ABS(SB(X1,Y1))+1)
910 IF DI AND (ST(X1,Y1)>0) THEN GOSUB 930
920 RETURN
930 REM druk veld af
940 IF INKEY$=" " THEN FOR WL=1 TO 500:NEXT WL
950 PLAY "o3s8m7014c"
960 GOTO 1540
970 END
980 REM zet computer
990 PRESET(15,170)
1000 IF AU THEN 1030

```

```

1010 COLOR 1:PRINT#1,"Mijn beurt ..."
1020 GOTO1050
1030 IF PL=1 THEN COLOR14:PRINT#1,"Speler aan zet
."
1040 IF PL=-1 THEN COLOR1:PRINT#1,"Computer aan z
et."
1050 BE=1000
1060 FOR TX=1 TO SI
1070 FOR TY=1 TO SI
1080 IF SGN(RB(TX, TY))=-SGN(PL) THEN 1150
1090 GOSUB 240
1100 X=TX:Y=TY:DI=0
1110 GOSUB670:REM expl.
1120 IF ET THEN MX=TX:MY=TY:GOTO 1170
1130 GOSUB1260:REM evaluatie
1140 IF (EN<BE) OR ((EN=BE)AND(RND(TIME)<.4)) THE
N BE=EN:MX=TX:MY=TY
1150 NEXT TY
1160 NEXT TX
1170 REM uiteindelijke zet
1180 PRESET(15,170):COLOR 5:PRINT#1,STRING$(20,21
9)
1190 GOSUB240
1200 X=MX:Y=MY:DI=-1
1210 CL=1:IF PL=1 THEN CL=14
1220 PRESET(15,170):COLOR CL:PRINT#1,"Dit is de z
et ...":RX=MX:RY=MY:GOSUB1840:PRESET(15,170):COLO
R 5:FORWL=1 TO 150:NEXTWL:PRINT#1,STRING$(20,219)
:CL=4:GOSUB1840
1230 GOSUB670:REM expl.
1240 GOSUB300
1250 RETURN
1260 REM evaluatie
1270 EN=0
1280 FOR X=1 TO SI
1290 FOR Y=1 TO SI
1300 EN=EN+SB(X,Y)
1310 IF SB(X,Y)<ST(X,Y)-1 THEN 1370
1320 EN=EN-2
1330 IF SB(X+1,Y)=ST(X+1,Y)-1 THEN EN=EN+10
1340 IF SB(X,Y+1)=ST(X,Y+1)-1 THEN EN=EN+10
1350 IF SB(X-1,Y)=ST(X-1,Y)-1 THEN EN=EN+10
1360 IF SB(X,Y-1)=ST(X,Y-1)-1 THEN EN=EN+10
1370 NEXT Y:NEXT X
1380 RETURN
1390 REM ini

```

```

1400 FOR X=1 TO SI
1410 FOR Y=1 TO SI
1420 CX(X,Y)=X0+28*(X-1)-20*(Y-1)
1430 CY(X,Y)=Y0-20*(Y-1)-14*(X-1)
1440 NEXT Y:NEXT X
1450 CLS:COLOR1,5,5:SCREEN2
1460 OPEN "GRP:" FOR OUTPUT AS #1
1470 FOR X=1 TO SI:FOR Y=1 TO SI
1480 X1=CX(X,Y)-5:Y1=CY(X,Y)+10
1490 PSET(X1,Y1),4
1500 DRAW"s4M+26,-13M-18,-18M-26,+13M+18,+18"
1510 NEXT Y:NEXT X
1520 IF AU THEN PRESET(15,10):PRINT#1,"D E M O
  Stop=<F1>"
1530 RETURN
1540 REM plaats stapel
1550 XX=CX(X1,Y1):YY=CY(X1,Y1)
1560 L=ABS(SQ(X1,Y1)):CL=5:GOSUB 1580:SQ(X1,Y1)=A
BS(SB(X1,Y1))
1570 L=ABS(SB(X1,Y1)):CL=-1*(PL=-1)-14*(PL=1):IF
SB(X1,Y1)=0 THEN RETURN
1580 PRESET(XX,YY)
1590 FOR N1=1 TO L
1600 CIRCLE STEP(0,0),4,5,,.8
1610 CIRCLE STEP(0,0),4,CL,3.14,6.28,.8
1620 COLOR CL
1630 DRAW"s4BM-4,0U2BM+8,+2U2BM-4,0"
1640 NEXT N1
1650 CIRCLE STEP(0,0),4,CL,,.8
1660 RETURN
1670 REM invoer zet speler
1680 PRESET(15,170):COLOR 14:PRINT#1,"Jouw beurt"
1690 CL=14:GOSUB1840
1700 A$=INKEY$:IF A$="" THEN 1700
1710 IF A$=CHR$(13) THEN1730
1720 IF PLAY(0) THEN 1720 ELSE PLAY "14o2s2m5000c
"
1730 MX=RX:MY=RY
1740 IF A$=CHR$(28) THEN MX=MX+1:IF MX>SI THEN 17
00 ELSE 1800
1750 IF A$=CHR$(29) THEN MX=MX-1:IF MX<1 THEN 170
0 ELSE 1800
1760 IF A$=CHR$(30) THEN MY=MY+1:IF MY>SI THEN 17
00 ELSE 1800
1770 IF A$=CHR$(31) THEN MY=MY-1:IF MY<1 THEN 170
0 ELSE 1800

```

```

1780 IF A$=CHR$(13) THEN PLAY"14s4m2500o6c":CL=4:
GOSUB1840:GOTO1820
1790 GOTO 1700
1800 CL=4:GOSUB1840:CL=14:RX=MX:RY=MY:GOSUB 1840
1810 GOTO1700
1820 PRESET(15,170):COLOR 5:FOR T=1 TO 300:NEXT T
:PRINT#1,STRING$(25,219)
1830 RETURN
1840 REM plaats cursor
1850 COLOR CL:PRESET(CX(RX,RY)-5,CY(RX,RY)+10)
1860 DRAW"C=CL;S4M+26,-13M-18,-18M-26,+13M+18,+18
"
1870 RETURN
1880 RUN

```

## 6.6 DO YOU MIND, MASTER?

DO YOU MIND MASTER is ongetwijfeld een superspel; het is eenvoudig en moeilijk tegelijk. Het spel zal zeker voor het nodige „aangenaam verpozen” zorgen.

De computer neemt 4 verschillende getallen tussen de 1 - 9 in gedachten en U dient die getallen dan te raden. Daartoe voert U steeds 4 getallen in een bepaalde volgorde in. De computer meldt ten eerste hoeveel getallen in het rijtje in ieder geval met de te raden getallen overeenstemmen. Ten tweede hoeveel getallen op de juiste plaats staan.

Na het RUN-commando kan het spel beginnen. In hoeveel beurten kunt u de combinatie vinden.

Voorbeeld:

DO YOU MIND, MASTER?

geef 4 getallen geraden juiste plaats

1 2 7 8	**	
3 5 1 7	***	
5 7 4 1	****	*
7 4 5 1	****	****

Geraden in 4 beurten!

Nog eens?



Programma:

```
10 REM *** DO YOU MIND MASTER ***
20 KEYOFF:SCREEN0:WIDTH40:COLOR15,1,1
30 KOP$="          DO YOU MIND, MASTER?"
                                     geef 4 ge
tallen  geraden  juiste plaats
                                     "
40 LOCATE0,0:PRINTKOP$
50 R=5:A=RND(-TIME):N=0
60 FORI=1TO4:A(I)=INT(RND(1)*9+1):A(I)=A(I)+(A(I)
=10)
70 NEXTI
80 FORK=1TO3:FORJ=K+1TO4:IFA(K)=A(J)THEN60
90 NEXTJ:NEXTK
100 KK=6:I=1:FORJ=1TO4:B(J)=-1:NEXTJ
110 LOCATEKK,R:A$=INPUT$(1):A=ASC(A$)
120 IF A=28 AND KK<12 THENKK=KK+2:I=I+1
130 IF A=29 AND KK>6 THENKK=KK-2:I=I-1
140 IFA=13 THEN170
150 IFA<48ORA>57THEN110
160 B(I)=VAL(A$):LOCATEKK,R:PRINTA$;:A=28:GOTO120
170 FORJ=1TO4:IFB(J)=-1THENI=J:KK=4+2*I:GOTO110
180 NEXTJ
190 N=N+1:G=0:P=0
200 FORK=1TO4:FORJ=1TO4
210 IFA(K)=B(J)THENG=G+1
220 NEXTJ
230 IFA(K)=B(K)THENP=P+1
240 NEXTK
250 LOCATE19,R:PRINTLEFT$(STRING$(4,"*"),G)
260 LOCATE30,R:PRINTLEFT$(STRING$(4,"*"),P)
270 GOSUB340
280 IFP<4THEN100
290 GOSUB340:GOSUB340
300 LOCATE3,R:PRINT"Geraden in";N;"beurten!"
310 LOCATE3,R+2:PRINT"Nog eens?";
320 A$=INPUT$(1):IF A$="N" OR A$="n"THEN END
330 RUN
340 IFR<19THENR=R+2:RETURN
350 LOCATE0,23:PRINT:LOCATE0,0:PRINTKOP$
360 LOCATE0,23:PRINT:LOCATE0,0:PRINTKOP$
370 RETURN
```

## 7 TEKSTVERWERKER EN DATABESTANDEN

Programma's voor tekstverwerking en het beheren van databestanden behoren stellig tot de meest geliefde programma's. Is het dan ook nog mogelijk deze twee programma's samen te voegen, dan wordt een bijzonder mooi geheel verkregen. De nu volgende programma's bieden u deze mogelijkheid.

### 7.1 TEKSTVERWERKER

Dit programma kent alle belangrijke functies nodig voor tekstverwerking en is daardoor zeer waardevol. Met behulp van dit programma kunnen we brieven opstellen, hierin verbeteringen aanbrengen, de tekst opslaan op cassette of diskette en afdrucken op een printer. Zelfs heeft het programma de zogenaamde „mail-merge“ faciliteit. Dit houdt in dat we het programma zelf diverse adressen of andere gegevens in de tekst kunnen laten invullen. Deze gegevens komen uit een bestand gemaakt met het databestandenprogramma uit paragraaf 7.2.

Nadat het programma is ingetypt en de checksums gecontroleerd (zie par 4.6) starten we de tekstverwerker met het commando:

```
RUN"TV.BAS <RETURN>
```

Na even gewacht te hebben verschijnt op het scherm:

**S**

KOPIEER	OPTIES	KIJK	ZOEK	SAVE
WIS	WIJZERS	PRINT	VERVANG	LOAD

Onderaan het scherm staan diverse commando's. Deze worden verderop behandeld. We zien dat linksboven op het scherm een geïnverteerde S staat. Dit noemen we de beginwijzer. Later komt aan de orde waar deze voor dient. Naast deze beginwijzer zien we de cursor. De cursor is de positie-aanwijzer op het scherm. We kunnen nu gaan typen. Aan het einde van een schermregel kunnen we gewoon doortypen. Het betreffende woord wordt meestal een beetje vreemd afgebroken, maar bij het afdrukken op de printer wordt dat wel gecorrigeerd.

Als we een alinea willen afsluiten drukken we op de RETURN-toets. De cursor gaat nu, in tegenstelling tot wat we zouden verwachten, niet naar de volgende regel. Op het scherm verschijnt een geïnverteerde P. Dit betekent dat de printer hier wél met een nieuwe alinea zal beginnen. We kunnen gewoon verder typen.

We moeten afraden om met de cursor een regel naar beneden te gaan, op de printer wordt het resultaat zelden fraai.

Een paar basisbewerkingen zijn het verwijderen en invoegen van tekst. We zullen beginnen met het verwijderen van tekst. Dit kan op twee manieren: ten eerste door gebruik van de toets met het opschrift BS (BackSpace). Het teken links van de cursor wordt nu verwijderd, de tekst rechts van de cursor zal aansluiten. Ten tweede door gebruik van de toets met het opschrift DEL (DElete). Hierdoor wordt het teken waarop de cursor staat gewist. Ook nu zal de tekst rechts van de cursor aansluiten.

Letters kunnen op de plaats van de cursor ingevoegd worden door op INS (INSert) te drukken. De cursor verandert nu van vorm en we kunnen tekst tussenvoegen. Het invoegen wordt uitgeschakeld door nogmaals op de INS-toets te drukken.

Een zeer belangrijk aspect van een tekstverwerker is wel de cursorbesturing. Met behulp van de cursortoetsen kunnen we de cursor in de vier richtingen verplaatsen:

- ↑ één teken naar boven
- ↓ één teken naar beneden
- één teken naar rechts
- ← één teken naar links

Dan zijn er nog twee mogelijkheden om de cursor te verplaatsen:

- HOME de cursor gaat naar het begin van de tekst
- CLS de huidige cursorpositie wordt de eerste positie op het scherm

## COMMANDO'S

Onderaan het scherm staan een tiental commando's. Deze kunnen we gebruiken door de bijbehorende functietoets in te drukken. Dit zijn:

- F1 Kopieer
- F2 Opties
- F3 Kijk
- F4 Zoek
- F5 Save
- F6 Wis
- F7 Wijzers
- F8 Print
- F9 Vervang
- F10 Load

Voor de laatste vijf commando's moet dus de SHIFT-toets tesamen met de betreffende functietoets worden ingedrukt.

Voor we deze commando's behandelen, zullen we eerst een en ander uitleggen omtrent de zogenaamde „wijzers”.

Er zijn twee wijzers, een **begin**- en een **eindwijzer**. Na het opstarten van deze tekstverwerker staat de beginwijzer op de allereerste positie van de tekst. Deze is zichtbaar als een geïnverteerde S (start). De eindwijzer staat op de allerlaatste positie van de tekst en is zichtbaar als een geïnverteerde E (einde). Deze E staat op de laatste positie die voor tekst gereserveerd is. De tekst die we kunnen intypen heeft dus een eindige lengte.

We kunnen de beginwijzer verplaatsen door de control-toets samen met de S toets in te drukken. De beginwijzer komt dan op de huidige cursorpositie te staan.

De eindwijzer kunnen we verplaatsen door de cursor op de gewenste positie te plaatsen en de control-toets samen met de E in te drukken. Het is niet mogelijk de beginwijzer na de eindwijzer te plaatsen.

Alle commando's die we kunnen geven met behulp van de functietoetsen hebben alleen betrekking op de tekst tussen de begin- en eindwijzer, met uitzondering van F5.

Deze wijzers kunnen teruggezet worden op hun oorspronkelijke positie door het indrukken van functietoets F7. Dus:

Control-S	verplaatsen beginwijzer
Control-E	verplaatsen eindwijzer

Hieronder volgt een overzicht van de werking van de verschillende commando's:

#### F1:Kopieer

De tekst tussen begin- en eindwijzer wordt tussengevoegd voor de cursor. De cursor mag zelf echter niet tussen de begin- en eindwijzer staan.

#### F2:Opties

Nadat we deze functietoets hebben ingedrukt, verschijnt het volgende op het scherm:

#### PRINTER

```
REGELBREEDTE : 80
PAPIERREGELS : 66
TEKST REGELS : 60
```

#### FILES

```
FILENAAM      : *DEFAULT  *
DEVICENAAM    : *CAS*
```

#### STRINGS

```
ZOEKSTRING    : *  *
VERVANG DOOR  : *  *
```

Verschillende waarden kunnen worden ingevoerd. Om tussen de verschillende regels te kunnen wisselen gebruiken we de cursor-omhoog en de cursor-omlaag toetsen. Om binnen een waarde iets te veranderen kunnen we gebruik maken van de cursor-links en cursor-rechts toetsen.

## REGELBREEDTE

Hiermee kunnen we de afdrukbreedte instellen. Het minimum aantal tekens per regel is 40, het maximum 250. Het invoeren van deze waarde gaat echter op een ietwat ongewone wijze. Met behulp van de cursor-links en cursor-rechts toetsen kunnen we deze waarde af laten tellen c.q. op laten lopen.

## PAPIERREGELS

Hiermee kunnen we het aantal regels dat op een printervel past aangeven. Minimaal is dit het aantal regels dat bij TEKSTREGELS is ingevoerd, maximaal 100. Het invoeren gaat op dezelfde wijze als bij REGELBREEDTE.

## TEKSTREGELS

Het aantal regels dat we op een printervel willen laten afdrukken kunnen we hiermee instellen. Dit is minimaal 20, maximaal het ingestelde aantal PAPIERREGELS. Het verschil tussen het aantal PAPIERREGELS en het aantal TEKSTREGELS wordt gelijkelijk verdeeld over het begin en het einde van het printervel. Het instellen gaat weer op dezelfde wijze als bij REGELBREEDTE.

## FILENAAM

Hier kunnen we de filenaam van de te laden of te saven file opgeven. Bij het laden van een file wordt de tekst in het geheugen gewist.

## DEVICENAAM

De naam van het apparaat waarop de file gesaved moet worden, of waarvan een file geladen moet worden. Er kan gekozen worden uit cassette (CAS) en diskette (A of B) en als uitzondering hierop de quickdisk (QD).

## ZOEKSTRING

Als een zoekcommando gegeven wordt moet naar de ingevulde tekst (string) gezocht worden. Indien een vervangcommando gegeven wordt is dit de (oude) tekst die door nieuwe tekst vervangen moet worden.

## VERVANG DOOR

Indien een vervangcommando gegeven wordt wordt de gezochte tekst (ZOEKSTRING) vervangen door de hier ingevulde tekst.

### **F3:Kijk**

Met dit commando kan bekeken worden hoe de tekst er straks op de printer zal gaan uitzien. Met de cursortoetsen kunnen we naar links, rechts en onder lopen. De cursor-omhoog toets zorgt ervoor dat het begin van de tekst weer in beeld komt. Door het geven van een RETURN verlaten we dit commando.

In de KLIK-mode kan de tekst alleen bekeken worden, dus niet gewijzigd.

### **F4:Zoek**

Zoek in het tekstdeel tussen de begin- en eindwijzer naar het gezochte woord. Dit woord moeten we eerst in de opties (F2) hebben ingevuld. Het eerste voorkomen van het gezochte woord wordt gevonden. Dit wordt bovenaan het scherm gezet. Als we het volgende voorkomen van het woord willen zoeken, dan moeten we eerst de beginwijzer verzetten. We verplaatsen de cursor één positie naar rechts en geven control-S. Als we nu opnieuw F4 geven wordt het volgende voorkomen van de gegeven tekst gezocht.

### **F5:Save**

De file wordt opgeslagen onder de bij OPTIES opgegeven naam en op het bij OPTIES opgegeven device (randapparaat). Vergeet niet eerst de cassetterecorder aan te zetten als daarop gesaved moet worden.

### **F6:Wis**

De tekst tussen begin- en eindwijzer wordt gewist. Dit kan niet ongedaan worden gemaakt.

### **F7:Wijzers**

Zet de begin- en eindwijzer terug op de posities die ze hadden direct na het opstarten. Dit zijn de eerste en laatste posities van de tekst.

### **F8:Print**

Het stuk tekst tussen begin- en eindwijzer wordt afgedrukt. Indien lettertekens worden gebruikt die met behulp van de CODE-toets gemaakt zijn, dan moet wel een MSX-printer gebruikt worden. Er wordt standaard altijd uitgevuld.

Indien aan het begin van een regel spaties staan worden deze gehandhaafd. Op deze manier is het mogelijk inspringingen te maken.

De printer-instellingen kunnen ingegeven worden bij de OPTIES (F2).

### **F9:Vervang**

De in OPTIES aangegeven tekst (oude string) wordt automatisch vervangen door de tekst achter VERVANG, eveneens in OPTIES. Dit geldt alleen tussen begin- en eindwijzer en alleen indien dit in het geheugen past. Dus als de nieuwe tekst langer is dan de oude tekst en de tekst is ongeveer maximaal van lengte, dan is het mogelijk dat het geheugen vol raakt.

### **F10:Load**

Een tekst die is opgeslagen wordt in het geheugen geladen. Eventueel aanwezige tekst gaat verloren.

## SAMENGESTELDE COMMANDO'S

Het is mogelijk door het geven van meer dan een commando een „nieuw“ commando te maken. Als voorbeeld: De hele tekst kan worden gewist door eerst WIJZERS en vervolgens WIS te geven.

Tekst kan verplaatst worden door begin- en eindwijzer om de betreffende tekst te zetten, de cursor op de bestemmingsplaats en achtereenvolgens KOPIEER en WIS te geven.

## CONTROL-KARAKTERS

Deze worden gemaakt door de control-toets samen met het betreffende karakter in te drukken. We hadden dit al gezien bij het verplaatsen van de begin- en eindwijzer.

Nog twee control-karakters zijn van belang:

control-B                   Paginascheiding. De huidige alinea eindigt hier en een nieuwe pagina wordt begonnen. Dit geldt alleen voor de printer.

control-D                   Zie onder MAIL/MERGE.

## ALGEMENE OPMERKINGEN

Natuurlijk is het zoals altijd van groot belang om regelmatig te saven.

Indien meer dan één toets tegelijk aangeslagen wordt zal het programma dat niet blokkeren, maar er iets in proberen te herkennen. Dit kan dus ook een WIS of LOAD opdracht zijn. Het gevolg hiervan is verlies van de tekst. Het geheel wissen van de tekst kan voorkomen worden door de begin- en eindwijzer direct na elkaar te plaatsen. Tegen een LOAD-actie kan niets gedaan worden.

Mocht er iets mis gaan, waardoor het programma per ongeluk verlaten wordt druk dan meteen op F5. Het programma wordt nu opnieuw gestart zonder dat er tekst verloren gaat. Gewoon herstarten, dus met verlies van tekst, kan door het intypen van RUN.

Indien de standaardinstellingen voor de printer en het device (randapparaat) niet overeenkomen met de standaardinstellingen van de aangesloten printer en het aangesloten device dan kunnen deze zo veranderd worden dat niet telkens het commando OPTIES gegeven hoeft te worden om dit aan te passen. In regel 20005 staan deze instellingen. Door wijziging van deze regel kunnen deze aangepast worden.

Als gebruik wordt gemaakt van een TV als beeldscherm, of van een monitor met geluid, horen we bij het maken van een fout een waarschuwingsgeluid. Ook elke toetsaanslag maakt geluid.

Beëindig het programma door de computer uit te schakelen.

Programma:

```
1 '*****
2 '*      MSX-TEKSTVERWERKER *
3 '*      VERSIE 1.1 1985 *
4 '*      H GROOT LIPMAN *
5 '*****
8 '
9 CLEAR 300,45000!:DEFINT A-Z:DEF FNH(X)=X\256+255:DEF FNL(X)
=X MOD 256-256*(X MOD 256 < 0)
10 GOTO 20000
15 REM INTYPEN VAN TEKST
20 D=0:IF PEEK(CK)=211 THEN D=1
23 IF PEEK(CK)=197 THEN IF PEEK(CK-1)<>211 THEN D=-1 ELSE IF
CK<TT THEN D=1 ELSE D=-2
25 IF D<>0 GOTO 183
26 A$=INKEY$:IF A$="" GOTO 20
30 A=ASC(A$):IF A<28 GOTO 200 ELSE IF A<32 GOTO 60 ELSE IF A=
127 GOTO 320 ELSE IF A>191 AND A<224 GOTO 20
40 IF IM THEN U=USR9(CK)
45 IF PEEK(CK)=211 OR PEEK(CK)=197 GOTO 20 ELSE POKE CK,A
50 GOTO 120
55 REM CURSORBEWEGINGEN
60 CD=STICK(0)
70 ON CD GOTO 100,110,120,130,140,150,160,170
80 GOTO 20
100 D=-40:GOTO 180
110 D=-39:GOTO 180
120 D= 1:GOTO 180
130 D= 41:GOTO 180
140 D= 40:GOTO 180
150 D= 39:GOTO 180
160 D=- 1:GOTO 180
170 D=-41
180 IF NOT PEEK(&HFBEB) AND 1 AND A<32 THEN D=D*10
183 CK=CK+D
185 IF CK<HK THEN IF HK-40>=TS THEN HK=HK-40:GOTO 185 ELSE HK
=TS: IF CK<HK THEN CK=HK ELSE ELSE IF CK>HK+799 THEN IF HK+83
9<=TT THEN HK=HK+40:GOTO 185 ELSE HK=TT-799: IF CK>HK+799 THE
N CK=HK+799
190 GOTO 500
200 REM CONTROLCODES
205 ON INSTR("€PBDHMKLR",CHR$(A+64)) GOTO 220,230,240,250,26
0,270,280,290,300,310
210 GOTO 20
220 IF CK>USR3(&HC5) THEN GOTO 20
222 SK=USR3(&HD3):U=USR1(SK)
224 IF SK<CK THEN CK=CK-1
226 A=211: GOTO 400
230 IF CK<=USR3(&HD3) THEN GOTO 20
232 EK=USR3(&HC5):U=USR1(EK)
234 IF EK<CK THEN CK=CK-1
235 IF CK<>TT AND PEEK(TT)=197 THEN POKE TT,32
236 A=197:GOTO 400
240 A=208: GOTO 400
```



```

250 A=194: GOTO 400
260 A=196: GOTO 400
270 IF CK=TS OR PEEK(CK-1)=211 OR PEEK(CK-1)=297 GOTO 20 ELSE
  CK=CK-1:GOTO 320
280 A=208:GOTO 400
290 CK=TS: HK=TS: GOTO 500
300 IF CK+800>TT THEN HK=TT-799:CK=HK ELSE HK=CK
305 GOTO 500
310 IF IM THEN IM=0:PRINTCHR$(27)+"x4";: ELSE IM=1:PRINTCHR$(
27)+"y4";
315 GOTO 500
320 IF PEEK(CK)=211 OR PEEK(CK)=197 GOTO 20 ELSE U=USR1(CK)
325 GOTO 500
400 U=USR9(CK):GOTO 45
500 REM AFDRUKKEN BEELDSCHERM
510 LOCATE,,0:U=USR0(HK)
520 D=CK-HK: CY=D\40: CX=D-CY*40:LOCATE CX,CY,1
530 GOTO20
7000 PRINTCHR$(7);:PRINTCHR$(7);:GOTO 8000
7900 GOSUB 23000
8000 REM RETURN VAN FUNCTIE
8010 FOR X=1 TO 10:KEY(X) ON:NEXT
8020 GOTO 500
9000 REM START EEN FUNKTIE
9005 LOCATE,,0:FOR X=1 TO 10:KEY(X) OFF:NEXT
9010 BK=USR3(&HD3):NS=USR4(0):EK=USR3(&HC5)
9020 IF EK=TT THEN SL=NS-BK ELSE SL=EK-BK-1
9030 POKE 56935!,FNL(BK):POKE 56936!,FNL(BK):POKE 56933!,FNL(
EK):POKE 56934!,FNL(EK):POKE 56925!,SL MOD 256:POKE 56926!,SL
\256
9040 RETURN
10000 REM FN1:KOPIEER
10010 GOSUB 9000
10020 IF CK>BK AND CK<=EK AND NS>=CK OR BK+1=EK OR CK+SL>TT T
HEN RETURN 7000
10060 U=USR8(CK)
10070 IF CK<BK THEN BK=BK+SL:POKE 56935!,FNL(BK):POKE 56936!,
FNL(BK)
10080 U=USR6(CK)
10090 RETURN 8000
11000 REM FN7:WIJZERS
11010 GOSUB 9000:U#=USR1(BK)+USR1(USR3(&HC5))+USR9(TS)
11020 POKE TS,211:POKE TT,197
11030 RETURN 8000
12000 REM FN3:KIJK
12010 GOSUB 9000:S1=1:S0=0:RS=1:GOTO 12300
12020 A=ASC(INPUT$(1)): IF A>27 THEN ON A-27 GOTO 12100,12200
,12010,12300
12030 IF A=13 THEN GOSUB 23000:RETURN 8000 ELSE GOTO 12010
12100 RS=RS+30:IF RS+30>PB THEN RS=PB-39
12110 S1=B1:S0=B0:MD=BM:CL=BC:US=BU
12120 GOTO 12500
12200 RS=RS-30:IF RS<1 THEN RS=1
12210 GOTO 12110
12300 IF S0=1 GOTO 12020 ELSE B1=S1:B2=S2:B0=S0:BM=MD:BC=CL:B
U=US

```

```

12500 X=22:CLS
12510 GOSUB 17500:PRINTMID$(R$,RS,40);
12520 IF S0=0 THEN X=X-1:IF X GOTO 12510
12530 GOTO 12020
13000 REM FN4:ZOEK
13010 GOSUB 9000:IF BK+1=EK THEN RETURN 7000
13030 A=USR7(VARPTR(Z$))-1:IF A>BK+SL THEN RETURN 7000
13040 CK=A:IF A>TT-799 THEN HK=TT-799 ELSE HK=CK
13050 RETURN 8000
14000 REM FN9:LAAD
14010 GOSUB 9000:POKE TS,211: POKE TT,197:EK=TT:NS=EK-1:BK=TS
:GOSUB 9020:U=USR2(BK):POKE TS+1,32:POKE TT,197
14020 A$=D$+"":'+F$
14030 IF LEN(D$)>1 THEN IF (ASC(D$) OR 32)=113 AND (ASC(MID$(
D$,2)) OR 32)=100 THEN _BLOAD(A$) ELSE BLOAD A$ ELSE BLOAD A$
14040 CK=TS:HK=TS:RETURN 8000
15000 REM FN6:DELETE
15010 GOSUB 9000:IF BK+1=EK OR NS<=BK THEN RETURN 8000
15020 U=USR2(BK)
15030 IF CK>BK THEN IF CK<EK THEN CK=BK ELSE CK=CK-EK+BK+1
15040 IF HK>CK THEN HK=CK
15050 RETURN 8000
16000 REM FN2:OPTIES
16005 GOSUB 9000:GOSUB 16000
16010 GOTO 16060
16020 IF A=13 GOTO 16800
16030 IF A=30 AND CSRLIN>3 OR A=31 AND CSRLIN<23 THEN PRINTCH
R$(A);
16060 LOCATE 20,CSRLIN:ON CSRLIN-2 GOSUB 16100,16110,16120,16
080,16080,16080,16080,16080,16080,16080,16170,16180,16080,160
80,16080,16080,16080,16080,16080,16290,16300
16070 GOTO 16020
16080 RETURN
16100 UB=250:LB=40:PR=PB:GOSUB 16500:PB=PR:RETURN
16110 UB=100:LB=RP:PR=LP:GOSUB 16500:LP=PR:RETURN
16120 UB=LP:LB=20:PR=RP:GOSUB 16500:RP=PR:RETURN
16170 IP$=F$:MP=29:GOSUB16700:F$=IP$:RETURN
16180 IP$=D$:MP=22:GOSUB16700:D$=IP$:RETURN
16290 IP$=Z$:MP=34:GOSUB16700:Z$=IP$:RETURN
16300 IP$=V$:MP=34:GOSUB16700:V$=IP$:RETURN
16500 REM FN7:OP- EN AFTELLEN
16510 A=ASC(INPUT$(1)):IF A=13 OR A=30 OR A=31 THEN RETURN
16520 IF A=28 AND PR<UB THEN PR=PR+1: LOCATE 19,CSRLIN:PRINT
PR;:LOCATE 20,CSRLIN
16530 IF A=29 AND PR>LB THEN PR=PR-1: LOCATE 19,CSRLIN:PRINT
PR;:LOCATE 20,CSRLIN
16540 GOTO 16500
16600 REM FN7:SPATIES VERWIJDEREN
16610 IF LEFT$(U$,1)=" " THEN U$=MID$(U$,2):GOTO 16610
16620 IF RIGHT$(U$,1)=" " THEN U$=LEFT$(U$,LEN(U$)-1):GOTO 166
20
16630 RETURN
16700 REM FN7:STRINGINVOER
16710 A=ASC(INPUT$(1)): IF A =13 OR A=30 OR A=31 THEN RETURN
16720 IF A=28 AND POS(0)<MP THEN LOCATE POS(0)+1,CSRLIN ELSE
IF A=29 AND POS(0)>20 THEN LOCATE POS(0)-1,CSRLIN

```

```

16725 IF A=127 THEN A=32
16730 IF A>31 AND (A<192 OR A>223) THEN MID$(IP$,POS(0)-19,1)
=CHR$(A):A=POS(0):LOCATE 20,CSRLIN:PRINTIP$;:LOCATE A,CSRLIN:
A=28:GOTO 16720
16740 GOTO 16700
16800 REM FN7:KLAAR
16802 U$=D$:GOSUB 16600:D$=U$
16804 U$=Z$:GOSUB 16600:Z$=U$
16806 U$=V$:GOSUB 16600:V$=U$
16810 U$=F$:GOSUB 16600:F$=U$
16820 U$=D$:GOSUB 16600:D$=U$
16830 U$=Z$:GOSUB 16600:Z$=U$
16840 U$=V$:GOSUB 16600:V$=U$
16850 CLS:GOSUB 23000:RETURN 8000
16900 REM FN7:TEKEN SCHERM
16905 RESTORE 16900:CLS:FOR X=1 TO 12:READ Y,Z,A$:LOCATE Y,Z:
PRINTA$;:NEXT
16910 DATA 14,1,PRINTER,4,3,"REGELBREEDTE :",4,4,"PAPIERREGEL
S :",4,5,"TEKST REGELS :",14,10,FILES,4,12,"FILENAAM      : *
*",4,13,"DEVICENAAM   : *   *",14,20,STRINGS,4,22,"Z
OEKSTRING           : *"
16915 DATA 4,23,"VERVANG DOOR : *",35,22,*,35,23,*
16920 LOCATE 19,3:PRINTPB
16921 LOCATE 19,4:PRINTLP
16922 LOCATE 19,5:PRINTRP
16923 LOCATE 20,12:PRINTF$
16924 LOCATE 20,13:PRINTD$
16925 LOCATE 20,22:PRINTZ$
16926 LOCATE 20,23:PRINTV$;
16930 F$=F$+STRING$(10-LEN(F$),32)
16932 D$=D$+STRING$(3-LEN(D$),32)
16934 Z$=Z$+STRING$(15-LEN(Z$),32)
16936 V$=V$+STRING$(15-LEN(V$),32)
16940 LOCATE 20,22,1:RETURN
17000 REM FN8:PRINT
17010 GOSUB 9000:S1=1:S0=0
17020 GOSUB 17500:LPRINT R$:IF S0=0 GOTO 17020
17030 RETURN 8000
17500 IF S1 THEN IF BK+1=EK THEN S0=1:RETURN ELSE SL=(LP-RP)\
2:TL=RP:EL=LP-TL-SL:S1=0:MD=1:CL=SL:POKE 56931!,PB:POKE 56932
!,0:US=BK+1
17510 R$=SPACE$(PB)
17520 Y=VARPTR(R$):POKE 56929!,PEEK(Y+1):POKE 56930!,PEEK(Y+2
)
17530 ON MD GOTO 17600,17700,17800,17900
17600 CL=CL-1:IF CL=0 THEN MD=2:CL=TL
17610 RETURN
17700 US=USR5(US)
17710 XZ=INSTR(R$,CHR$(&HC5)):IF XZ THEN R$=LEFT$(R$,XZ-1)+SP
ACE$(PB-XZ+1):S0=1
17715 XZ=INSTR(R$,CHR$(&HD0)):IF XZ THEN R$=LEFT$(R$,XZ-1)+SP
ACE$(PB-XZ+1)
17720 XZ=INSTR(R$,CHR$(&HC2)):IF XZ THEN R$=LEFT$(R$,XZ-1)+SP
ACE$(PB-XZ+1):MD=3
17725 IF US>NS THEN S0=1:RETURN
17730 CL=CL-1:IF CL=0 THEN MD=4:CL=EL

```

```

17740 RETURN
17800 CL=CL-1:IF CL=0 THEN MD=4:CL=EL
17810 RETURN
17900 CL=CL-1:IF CL=0 THEN MD=1:CL=SL
17910 RETURN
18000 REM FN9:VERVANG
18010 GOSUB 9000:IF BK+1=EK THEN RETURN 7000
18030 FS=TT-NS+(EK=TT)
18040 DS=LEN(V$)-LEN(Z$):FV=0
18050 IF DS>FS THEN RETURN 7000
18055 SL=USR3(&HC5)-BK-1:IF SL<=0 THEN RETURN 7000
18057 POKE 56925!,SL MOD 256:POKE 56926!,SL\256
18060 A=USR7(VARPTR(Z$))-1:IF A>BK+SL THEN IF FV=1 THEN RETURN 8000 ELSE RETURN 7000
18065 POKE &HDE5D,ABS(DS):POKE &HDE5E,0
18070 IF DS>0 THEN U=USR8(A) ELSE IF DS THEN U=USR2(A-1)
18080 IF LEN(V$) THEN FOR X=1 TO LEN(V$):POKE A+X-1,ASC(MID$(V$,X)):NEXT:FV=1
18090 BK=A+LEN(V$)-1:POKE 56935!,FNL(BK):POKE 56936!,FNH(BK):FS=FS-DS
18100 GOTO 18050
19000 REM FN5:LAAD
19010 GOSUB 9000:U#=USR1(BK)+USR1(USR3(&HC5))+USR9(TS):POKE TS,211:POKE TT,197:GOSUB 9000:A$=D$+"":+F$
19020 IF LEN(D$)>1 THEN IF (ASC(D$) OR 32)=113 AND (ASC(MID$(D$,2)) OR 32)=100 THEN _BSAVE(A$,TS,NS):RETURN 8000
19030 BSAVE A$,TS,NS:RETURN 8000
20000 REM INIT
20005 KEY OFF:SCREEN 0:WIDTH 40:LOCATE10,10:PRINT"EVEN GEDULD AUB.":LOCATE12,12:PRINT"*****";:PB=80:LP=66:RP=60:F$="DEFAULT":D$="CAS":TS=&HAF8:TT=&HDAA8:CK=TS+1:IM=0:
CX=0:CY=0:HK=TS
20010 GOSUB 22000
20015 POKE 56941!,256+TS MOD 256:POKE 56942!,255+TS\256:POKE 56939!,256+TT MOD 256:POKE 56940!,255+TT\256
20020 POKE TS,211:POKE TT,197:EK=TT:NS=EK-1:BK=TS:GOSUB 9020:U=USR2(BK):POKE TS+1,32:POKE TT,197
20030 ON KEY GOSUB 10010,16005,12010,13010,19010,15010,11010,17010,18010,14010
20040 FOR X=1 TO 10:KEY X,"":KEY(X) ON: NEXT
20050 KEY5,CHR$(12)+"GOTO 7900"+CHR$(13)
20060 GOSUB 23000:GOTO 500
21000 REM INIT:MAAK CONTROLKARAKTERS
21010 FOR X=BASE(2)+1536 TO BASE(2)+1743:VPOKE X,255-VPEEK(X-1024):NEXT:RETURN
22000 REM INIT:DUMP MACHINECODE
22005 SUM=0:RESTORE 22000
22010 READ A$:IF A$="s" OR A$="S" THEN IF SUM=178 THEN GOTO 22100 ELSE PRINT "DATA ERROR":STOP
22020 Y=VAL("&H"+A$):IF LEN(A$)=4 THEN X=Y:PRINTCHR$(8)+" "+CHR$(8);: ELSE POKE X,Y: X=X+1:SUM=SUM XOR Y
22030 GOTO 22010
22100 DEFUSR0=56657!:DEFUSR1=56857!:DEFUSR2=56757!:DEFUSR3=56717!:DEFUSR4=56677!:DEFUSR5=56307!:DEFUSR6=56631!:DEFUSR7=56557!:DEFUSR8=56228!:DEFUSR9=56180!
22110 RETURN

```

```

22200 REM 0:AFDRUKKEN
22210 DATA DD51, 2A, F8, F7, 11, 00, 00, 01, 20, 03, C3, 5C, 00
22220 REM 1:DELETE 1 KARAKTER
22230 DATA DE19, 2A, 6B, DE, ED, 5B, F8, F7, B7, ED, 52, 20, 06, 1A, FE, C5,
C8, 18, 0E, E5, C1, 2A, F8, F7, 23, ED, B0, 1A, FE, C5, 20, 01, 1B, 3E, 20, 12, C
9
22240 REM 2:DELETE MEERDERE KARAKTERS
22250 DATA DDB5, 2A, F8, F7, ED, 4B, 5D, DE, 09, 23, E5, 2A, 6B, DE, D1, D5,
B7, ED, 52, E5, C1, 03, E1, E5, ED, 5B, F8, F7, 13, ED, B0, E1, 2B, 2B, ED, 5B, F
8, F7, ED, 52, E5, C1, ED, 5B, 6B, DE, 1B, 2A, 6B, DE, 36, 20, 78, B1, C8, ED, B8
, C9
22255 DATA ED, B0, E1, 2B, 7E, FE, C5, C0, 36, 20, 2A, 6B, E2, 36, C5, C9
22260 REM 3:EERSTE VOORKOMEN CODE X
22270 DATA DD8D, 3A, F8, F7, 2A, 6B, DE, ED, 4B, 6D, DE, ED, 42, E5, C1, 03,
2A, 6D, DE, ED, B1, 2B, 22, F8, F7, C9
22280 REM 4:ZOEK TEKSTEINDE
22290 DATA DD65, 2A, 6B, DE, 7E, FE, 20, 28, 0B, FE, C5, 20, 0A, 36, 20, ED,
5B, 6B, DE, 12, 2B, 18, ED, 22, F8, F7, C9
22300 REM 5:UITVULLEN
22310 DATA DBF3, 2A, F8, F7, ED, 5B, 61, DE, 3A, 63, DE, 47, 7E, FE, 20, 20,
07, 23, 7E, FE, 20, 20, 01, 2B, 7E, E5, D5, E1, 77, E1, 13, 23, FE, D0, 28, 08, F
E, C2, 28, 04, 10, EE, 18, 04, 22, F8, F7, C9
22320 DATA E5, DD, E1, E5, 3E, 20, ED, 4B, 63, DE, 2A, 61, DE, ED, B1, E1, 20
, EA, DD, BE, 00, 20, 0B, 1B, D5, E1, BE, 28, 13, DD, 22, F8, F7, C9, 1B, D5, E1,
BE, 28, 08, 36, 20, DD, 2B, 2B, BE, 20, F8, DD, 22, F8, F7, 2A, 61, DE, 3E, 20, B
E, 20, 03, 23, 18, FA
22330 DATA BE, 28, 03, 23, 18, FA, BE, 20, 03, 23, 18, FA, E5, C1, ED, 5B, 63
, DE, 2A, 61, DE, 19, 2B, B7, ED, 42, D8, 09, E5, BE, 20, 03, 2B, 18, FA, ED, 5B,
61, DE
22340 DATA 1A, FE, 20, 20, 03, 13, 18, F8, 1A, FE, 20, 28, 03, 13, 18, F8, E5
, B7, ED, 52, E1, 38, E5, D5, E5, ED, 52, E5, C1, 03, E1, E5, E5, D1, 13, ED, B8,
E1, 23, D1, C1, E5, B7, ED, 42, E1, C8, C5, 18, CE
22350 REM 6:KOPIEER
22360 DATA DD37, ED, 4B, 5D, DE, 2A, 67, DE, 23, ED, 5B, F8, F7, ED, B0, C9
22370 REM 7:ZOEK
22380 DATA DCED, 2A, F8, F7, 46, 23, 5E, 23, 56, EB, 7E, D9, 2A, 67, DE, 23,
ED, 4B, 5D, DE, 03, ED, B1, 20, 13, E5, D9, D1, 1B, E5, C5, F5, 1A, BE, 13, 23, 2
0, 0A, 10, F8, E1, E1, E1, D9, 22, F8, F7, C9, F1, C1, E1, D9, 18, DF
22390 REM 8:INSERT MEERDERE KARAKTERS
22400 DATA DBA4, ED, 5B, 6B, DE, 2A, 6B, DE, ED, 4B, F8, F7, B7, ED, 42, 28,
06, ED, 4B, 5D, DE, ED, 42, E5, C1, 03, C5, 2A, 6B, DE, ED, 4B, 5D, DE, ED, 42, C
1, 1A, FE, C5, 20, 03, 0B, 1B, 2B, 78, B1, C8, ED, B8, C9
22410 REM 9:INSERT 1 KARAKTER
22420 DATA DB74, 2A, 6B, DE, ED, 4B, F8, F7, B7, ED, 42, C8, E5, C1, 2A, 6B,
DE, ED, 5B, 6B, DE, 2B, 1A, FE, C5, 20, 06, 7E, FE, D3, C8, 36, C5, ED, B8, 3E, 2
0, 12, C9
22999 DATA S
23000 REM TEKEN FUNCTIEBALK
23005 KEY OFF: A$=CHR$(1)+CHR$(86):LOCATE 0, 20:PRINTCHR$(1)+CH
R$(88);
23010 FOR X=1 TO 5:FOR Y=1 TO 7:PRINTCHR$(1)+CHR$(87);:NEXT:P
RINTCHR$(1)+CHR$(82);:NEXT:PRINTCHR$(8)+CHR$(8)+CHR$(1)+CHR$(
89);
23020 PRINTA$+"KOPIEER"+A$+"OPTIES "+A$+" KIJK "+A$+" ZOEK
"+A$+" SAVE "+A$+CHR$(1)+CHR$(84);

```

```

23030 FOR X=1 TO 5: FOR Y=1 TO 7:PRINTCHR$(1)+CHR$(87);:NEXT:
PRINTCHR$(1)+CHR$(85);:NEXT:PRINTCHR$(8)+CHR$(8)+CHR$(1)+CHR$
(83);
23040 PRINTA$+" WIS "+A$+"WIJZERS"+A$+" PRINT "+A$+"VERVANG
"+A$+" LOAD ";
23050 LOCATE0,0:VPOKE 959,22:RETURN
33187 IF PEEK(CK)=211 THEN CK=CK+1
33188 IF PEEK(CK)=197 THEN CK=CK-1: IF PEEK(CK)=211 THEN IF C
K>TS THEN CK=CK-1 ELSE CK=CK+2
50000 FORX=56180!TO56900!
50010 IF PEEK(X)=&HE2 THEN PRINTX
50020 NEXT:STOP

```

*Checksums:*

Gebruik het programma van paragraaf 4.6.

1:2FE1	2:062A	3:F3D0	4:3C84	5:AFE1	
8:0097	9:B3E4	10:B751	15:F21F	20:46CB	
23:C7C4	25:E2DA	26:9EF0	30:9C72	40:5A9E	
45:6304				blok-checksum :	2C08
50:71B0	55:0EF6	60:0058	70:C648	80:3A40	
100:35EE	110:7CEE	120:BCC8	130:8ED5	140:ADD5	
150:D4D5	160:E0D5	170:33B5	180:DFAB	183:D54E	
185:F7FF				blok-checksum :	4831
190:D7BC	200:83A7	205:920B	210:68C0	220:7B3C	
222:C2AE	224:BC63	226:D4EB	230:C8CF	232:BD76	
234:BC64	235:2301	236:26EC	240:99EB	250:E3EB	
260:25EC				blok-checksum :	0C45
270:3679	280:99EC	290:163A	300:2FE5	305:D53C	
310:CEAC	315:D63C	320:8E3B	325:D73C	400:B1A4	
500:590F	510:39A9	520:79DD	530:6CC0	7000:8201	
7900:DA15				blok-checksum :	0ECC
8000:2C2D	8010:3A83	8020:AEBD	9000:D916	9005:DED4	
9010:B726	9020:9DD7	9030:ED77	9040:DB76	10000:2163	
10010:5924	10020:70DF	10060:E723	10070:34D2	10080:F703	
10090:0289				blok-checksum :	E714
11000:29E8	11010:74FD	11020:3F91	11030:F288	12000:E40C	
12010:6AB1	12020:AAD8	12030:1DD7	12100:5688	12110:A4DF	
12120:8370	12200:F65B	12210:8762	12300:95F0	12500:4C6D	
12510:B3CA				blok-checksum :	3DE4
12520:97DD	12530:A760	13000:24B3	13010:3EFD	13030:708C	
13040:F6BB	13050:4289	14000:649C	14010:6943	14020:BD6F	
14030:8158	14040:EDE1	15000:917B	15010:0FCD	15020:66C0	
15030:A3BD				blok-checksum :	5DF7
15040:E3F4	15050:8289	16000:91CC	16005:382F	16010:B788	
16020:B8BC	16030:5105	16060:DB27	16070:CF80	16080:9B77	
16100:19F4	16110:3F2B	16120:C706	16170:5D63	16180:2FA1	
16290:9878				blok-checksum :	1F2F
16300:56B4	16500:A841	16510:3BF8	16520:82C6	16530:9A2E	
16540:EB90	16600:F4D2	16610:49B1	16620:8324	16630:9EF7	
16700:544D	16710:3BF9	16720:F497	16725:2751	16730:05B5	
16740:FB98				blok-checksum :	52EF
16800:C94F	16802:3588	16804:37C0	16806:37F0	16810:35B0	

16820:3608	16830:3840	16840:3870	16850:64CF	16900:87D7	
16905:9699	16910:314D	16915:375B	16920:E57C	16921:E686	
16922:E796				blok-checksum :	8701
16923:7C6D	16924:7D71	16925:7EA5	16926:FF85	16930:10D0	
16932:FE8C	16934:59D1	16936:54D1	16940:DAE0	17000:4A4A	
17010:8FA7	17020:AA49	17030:B289	17500:446B	17510:2A1B	
17520:4D61				blok-checksum :	04EC
17530:FBF9	17600:3D8F	17610:9FF7	17700:2767	17710:A26E	
17715:2CE0	17720:A30A	17725:0546	17730:DDF1	17740:A277	
17800:BDF1	17810:A1F7	17900:FD4D	17910:A2F7	18000:2B72	
18010:3F4D				blok-checksum :	B0EC
18030:1A48	18040:A29D	18050:28A8	18055:459E	18057:200F	
18060:844C	18065:3275	18070:4D63	18080:ECC1	18090:88E7	
18100:DB96	19000:A41D	19010:951E	19020:D64D	19030:4FA4	
20000:1E59				blok-checksum :	A980
20005:2DDA	20010:9992	20015:3479	20020:D75D	20030:9BBA	
20040:3C6C	20050:7A30	20060:62D6	21000:88C1	21010:E77B	
22000:580C	22005:61F3	22010:4910	22020:9AD0	22030:9F6E	
22100:0861				blok-checksum :	3F71
22110:94F7	22200:B680	22210:A248	22220:4103	22230:D55C	
22240:7BD6	22250:7F56	22255:960C	22260:9D69	22270:5052	
22280:7179	22290:C119	22300:E48D	22310:AB5F	22320:1795	
22330:7956				blok-checksum :	FA14
22340:9687	22350:173E	22360:4086	22370:CB2D	22380:D51C	
22390:64BD	22400:8DD1	22410:276E	22420:014F	22999:D04D	
23000:EA93	23005:642C	23010:F3EA	23020:D548	23030:B3E3	
23040:FF6A				blok-checksum :	71E6
23050:8C09	33187:FCB1	33188:F6E5	50000:A719	50010:9423	
50020:E182				blok-checksum :	D8F1

## 7.2 DATABESTANDEN

Een zeer nuttige toepassing van uw MSX is het beheren van databestanden. Denk bijvoorbeeld aan de gegevens naam, straatnaam en woonplaats. Deze gegevens vormen een adres-databestand.

Met behulp van het in deze paragraaf beschreven programma kunnen we zo'n databestand opslaan in de computer en er dan allerlei bewerkingen op uitvoeren, zoals gegevens opvragen, sorteren, afdrukken etc.

De listing van het programma is erg lang, maar het is zeker de moeite waard zoals zal blijken uit de vele mogelijkheden die het programma biedt.

In het nu volgende zal worden uitgelegd hoe met dit programma een compleet adressenbestand kan worden opgezet.

### BESTANDSDEFINITIE

Na het opstarten van het programma met het RUN-commando verschijnt een scherm met de volgende keuzemogelijkheden:

```
***** DATABESTAND *****  
  
niet gesorteerd                Vrij : 0  
  
Menu :  
  
    Nieuw      : Maak een nieuw datablok  
    Insert    : Voeg records toe  
    Verander   : Verander veldgegevens  
    Type      : Print op het scherm  
    Print     : Print op de printer  
    Rangschik : Sorteert het bestand  
    Save      : Save op diskette  
    Load     : Load van diskette  
    Kill      : Wis een file van disk  
    Catalog   : Druk alle files af  
    Menu      : Druk dit menu af  
    Exit      : Beeindig dit programma
```

Opdracht :

Door het intoetsen van de beginletter, of een willekeurige afkorting van elk eerste woord, kunnen we een bepaalde opdracht kiezen. Bovenin het scherm staat Vrij: xx, waarbij xx dan een getal is. Dit getal geeft aan hoeveel velden er nog vrij zijn.

We kiezen eerst de opdracht Nieuw : Maak een nieuw datablok. Dit doen we door de letter N in te typen. Het programma komt terug met de vraag:

Aantal velden?





## TOEVOEGEN VAN GEGEVENS (I)

Elke keer als het programma vraagt „Opdracht?“ kunnen we een van de opdrachten uit het hoofdmenu uitvoeren. Willen we het hoofdmenu zelf zien, geef dan een „M“ of gewoon RETURN.

We gaan gegevens invoeren en kiezen daarom voor de opdracht „Voeg records toe“ (letter I). Het programma vraagt één voor één naar de gegevens. We voeren bijvoorbeeld het volgende adres in:

```
***** Insert *****  
  
Record : 1                Pagina : 1  
  
Naam : Jansen  
Voornaam : Piet  
Straat : De Klerckstraat 128  
Postcode : 7502 FP  
Plaats : ZWOLLE  
Telefoon ? 038-13268
```

Als we willen ophouden met het toevoegen van records dan moeten we op functietoets F1 drukken. Het hoofdmenu komt dan weer in beeld.

Opmerkingen:

- Is een bepaald gegeven onbekend, dan niets invullen maar gewoon op RETURN drukken.
- Gegevens mogen alle lettertekens en cijfers bevatten. Een gegeven mag echter geen haakjes () of de tekens <> = bevatten. Ook kan beter geen gebruik gemaakt worden van de leestekens : en ;.

Op dezelfde wijze voeren we ook de volgende twee adressen in:

Jan Klaassen  
Wegtersweg 12  
6805 DB ARNHEM  
085-73215

Piet Peters  
Voorbeeldstraat 5  
7508 AA ZWOLLE  
038-14738

### **Belangrijk:**

De geheugencapaciteit van de computer is uiteraard beperkt en dus kan het databestand vol raken. Hoeveel adressen het bestand kan bevatten is niet te voorspellen omdat dat afhankelijk is van de lengte van namen en adressen. Als het bestand vol is verschijnt de mededeling:

Database is vol

Er kunnen maximaal 800 velden gebruikt worden. Bij MSX-I kunnen deze ongeveer 5000 tekens bevatten, bij MSX-II ongeveer 4000.

## GEGEVENS OPVRAGEN (t)

Het hebben van een databestand is alleen zinvol als het mogelijk is snel daaruit bepaalde gegevens op te vragen. Dit programma beschikt over uitgebreide selectiemogelijkheden. Kies de opdracht „Type : Print op het scherm” (letter T).

Achter de ingetypte T vullen we in welke velden we willen zien. Daarbij hebben we de volgende mogelijkheden:

1. Veldnamen mogen worden afgekort zolang ze maar van elkaar te onderscheiden zijn.
2. Wordt tussen de veldnamen een spatie ingetoetst, dan worden de gegevens onder elkaar weergegeven.
3. Wordt voor de veldnaam een puntkomma ingetoetst, dan wordt het gegeven achter het voorgaande veld weergegeven. Wordt voor de eerste veldnaam een puntkomma getypt dan wordt tussen de verschillende records geen regel overgeslagen.
4. Wordt voor een veldnaam een dubbele punt (:) ingetoetst, dan wordt ook de veldnaam zelf getoond.
5. Wordt „NO” ingetypt, dan wordt ook het recordnummer weergegeven.
6. Staat er in plaats van het eerste woord een cijfer, dan worden alleen de gegevens van het record met dat nummer getoond.
7. Typen we niets in, maar drukken we alleen op de RETURN-toets, dan wordt het hele bestand getoond.
8. Wordt een bereik meegegeven, bijvoorbeeld T 1-20 dan worden alle records van 1 tot 20 getoond.

Het weergeven van de records stopt zodra het scherm vol is. Met de cursor-omlaag toets wordt het volgende scherm getoond, met de cursor-omhoog toets de vorige records.

### *Voorbeeld:*

T vo;na stra pla

Behalve opgeven welke velden we willen zien kunnen we ook een selectiecriteria opgeven. Dat geven we aan met het woord „als” gevolgd door een voorwaarde. De volgende voorwaarden zijn mogelijk:

veldnaam = gegeven (gelijk aan)  
veldnaam > gegeven (groter dan)  
veldnaam < gegeven (kleiner dan)  
veldnaam >= gegeven (groter gelijk)  
veldnaam <= gegeven (kleiner gelijk)  
veldnaam <> gegeven (ongelijk)

Het is ook mogelijk op een samenstelsel van voorwaarden te testen door tussen de voorwaarden het woordje „en” of het woordje „of” te plaatsen. Bovendien is het toegestaan voorwaarden tussen haakjes (tot 11 niveaus diep) te plaatsen waardoor zeer ingewikkelde tests mogelijk zijn.

### *Opmerkingen:*

Instructies als: NO, ALS, OF en EN mogen zowel in hoofd- als in kleine letters worden ingevoerd.

Als een veldnaam of testvoorwaarde spaties bevat dient deze tussen aanhalingstekens gezet te worden. Bijvoorbeeld: veld 1 wordt dan opgegeven als "veld 1". Dus:

```
t als "veld 1">a
```

**Voorbeeld:**

```
T vo;na als pl = ZWOLLE
```

Toegestaan is:

```
T voornaam naam straat  
T ;vo na str  
T :no voornaam;naam tel  
T RETURN  
T 2 voornaam;naam  
T als naam = Jansen  
T als naam> = J en naam<K) of plaats = ZWOLLE  
T no voornaam;naam straat plaats tel als plaats = ZWOLLE  
T. Peters  
T. n;s Peters (het bestand moet dan wel op het veld naam gesorteerd zijn)
```

De volgende functietoetsen zijn van belang:

```
F1  terug naar hoofdmenu  
F2  stop niet na elke volle schermpagina. Elke andere toets schakelt dit weer uit.  
F6  start opnieuw bij eerste pagina  
↓   volgende pagina  
↑   vorige pagina
```

## AFDRUKKEN GEGEVENS OP PRINTER (P)

Dit programma biedt ook de mogelijkheid gegevens af te drukken op de printer, bijvoorbeeld voor het maken van etiketten. Kies daarvoor opdracht **P** uit het hoofdmenu. We kunnen nu op dezelfde manier selectievoorwaarden opgeven als beschreven bij Type.

**Voorbeeld:**

```
P voornaam;naam straat postcode;plaats als naam> = K
```

```
Jan Klaassen  
Wegtersweg 12  
6805 DB ARNHEM
```

```
Piet Peters  
Voorbeeldstraat 5  
7508 AA ZWOLLE
```

## WIJZIGEN VAN GEGEVENS (V)

Met de opdracht V (verander veldgegevens) kunnen we gegevens wijzigen. Achter de V kunnen we weer, net als bij Type, voorwaarden invullen waaraan het record moet voldoen dat we willen wijzigen. Het betreffende record wordt opgezocht en op het scherm getoond. Met behulp van F4 en F5 kunnen we naar het vorige resp. volgende record gaan. Met behulp van F1 kunnen we terug naar het hoofdmenu.

Alleen records en velden die binnen het opgegeven bereik liggen kunnen worden veranderd. In record 0 staan de veldnamen. De veldnamen kunnen gewijzigd worden met het commando V 0.

### Voorbeeld:

V straat plaats als na = Peters

Op het scherm verschijnt nu:

```
***** Verander *****  
  
Record : 3                Pagina : 1  
  
Straat ? Voorbeeldstraat 5  
Plaats : ZWOLLE
```

Alleen de straat en plaats van het record dat Peters bevat kunnen worden veranderd. Geen andere velden van dit record of dezelfde van een ander record kunnen nu gewijzigd worden.

Als we niets opgeven achter V, dan kunnen we alle records en velden wijzigen. Met de functietoetsen F4 en F5 kunnen we weer de vorige en volgende records bekijken.

Nadat we het commando gegeven hebben wordt de cursor aan het begin van het eerste veld geplaatst dat veranderd kan worden. De volgende acties zijn nu mogelijk:

- cursor naar rechts
- ← cursor naar links
- DEL teken waarop de cursor staat wordt verwijderd
- BS teken voor de cursor wordt verwijderd
- INS er kan tekst ingevoegd worden op de cursorpositie. Het invoegen wordt beëindigd door elke andere instructie of cursorverplaatsing en ook door het nogmaals indrukken van de INS-toets.
- ↑ vorig veld. Als er meer pagina's zijn en de cursor op het eerste veld van de pagina staat, wordt automatisch de vorige pagina afgedrukt. De cursor staat dan weer op het eerste veld van die pagina.
- ↓ volgend veld. Als er meer pagina's zijn en de cursor staat op het laatste veld van de pagina, wordt automatisch de volgende pagina afgedrukt. De cursor komt op het eerste veld van deze pagina te staan.

Verder kan nog van een aantal functietoetsen gebruik worden gemaakt:

- F1 terug naar hoofdmenu
- F3 wis alles dat achter de cursor staat en geef RETURN

- F4 vorig record
- F5 volgend record
- F8 wis alles dat achter de cursor staat
- F10 wis het gehele record (er wordt om bevestiging gevraagd)

Is de tekst in een veld te lang dan wordt automatisch gescrolled.

Na beëindiging van het veranderen wordt het bestand automatisch weer gesorteerd. Dit natuurlijk alleen als het bestand voordat we gingen wijzigen ook gesorteerd was.

**Voorbeeld:**

V als naam = Peters

**WISSEN VAN RECORDS**

Een record kan gewist worden door vanuit de Verander-opdracht functietoets F10 te gebruiken. Zie hiervoor verder bij Veranderen.

**SORTEREN VAN GEGEVENS (R)**

Door het geven van de opdracht **R** gevolgd door een **veldnaam**, wordt het bestand gesorteerd op het opgegeven veld. Op elk veld kan gesorteerd worden. Voor numeriek sorteren moet echter bij de bestandsdefiniëring een punt voor de veldnaam zijn opgegeven. Zie hiervoor BESTANDSDEFINITIE.

Indien na de R een nul (0) opgegeven wordt, blijft het bestand ongesorteerd. Het is weer toegestaan de veldnaam af te korten.

**SAVEN VAN GEGEVENS (S)**

Met behulp van deze opdracht is het mogelijk gegevens op te slaan op diskette. Op dezelfde wijze als bij Type is het mogelijk voorwaarden te geven waaraan het te save bestand moet voldoen. Dit is vooral handig om bestandsdelen te selecteren voor gebruik met Mail/Merge. Moet het gehele bestand gesaved worden dan moeten geen voorwaarden gegeven worden. De Save-opdracht heeft dus de volgende vorm:

**S d:filenaam voorwaarden**

De **S** staat voor Save, de **d**: voor de drive-aanduiding. Deze is optioneel en mag dus weggelaten worden. De filenaam mag maximaal 8 tekens lang zijn. Wordt geen extensie opgegeven dan wordt de file onder de opgegeven naam met extensie .DAT weggeschreven. Indien nodig wordt onder dezelfde naam een backup file aangemaakt die de extensie .DAB krijgt. Alle bij Type gegeven voorwaarden mogen bij Save gebruikt worden.

**Opmerkingen:**

- Alle files krijgen de extensie .DAT
- Backup files worden door het programma aangemaakt en krijgen de extensie .DAB.

### **Voorbeelden:**

S adressen

Alle gegevens van alle adressen worden opgeslagen

S adressen v;n str po;pl als pl = Zwolle

Voornaam, naam, straat, postcode en plaats worden opgeslagen van alle personen die in Zwolle wonen.

### **LADEN VAN GEGEVENS (L)**

De gegevens die met Save zijn opgeslagen kunnen met deze opdracht weer geladen worden.

De manier waarop dit commando gegeven moet worden is als volgt:

#### **L d:filenaam**

Ook nu is de drive-aanduiding (d:) optioneel. Bij de filenaam hoeft geen extensie opgegeven te worden, het mag echter wel.

Let wel, een eventueel in het geheugen aanwezig bestand wordt gewist.

### **OPVRAGEN DIRECTORY (C)**

Een overzicht van de files op de in de drive aanwezige diskette wordt gegeven.

De manier waarop dit commando moet worden gegeven is:

#### **C d:naam**

Waarbij d: de drive aangeeft, dit mag weggelaten worden, er kan gekozen worden tussen A: en B:.. Voor „naam” kan een filenaam ingevuld worden, dit hoeft echter niet. Wel is het toegestaan met zogenaamde wildcards te werken. Een wildcard is een teken dat een willekeurig ander teken, of groep tekens, kan vervangen. Zo betekent bijvoorbeeld:

- \* Alle tekens in de filenaam, ongeacht welke, of alle tekens in de extensie. Voorbeeld: \*.dat, hieraan voldoen alle files met de extensie .dat ongeacht de filenaam.
- ? Vervangt een enkel willekeurig teken in de filenaam of extensie. Voorbeeld: bestand?.dab, hieraan voldoen o.m.: bestand1.dab, bestand5.dab en bestanda.dab.

Als helemaal niets wordt opgegeven worden alleen files met de extensie .DAT getoond. Wilen we een overzicht van alle files op de diskette dan moeten we \*. \* opgeven.

### **VERWIJDEREN VAN FILES (K)**

Het is mogelijk een file van diskette te verwijderen, dit gaat door middel van de K(ill) opdracht. De manier van invoeren van dit commando is weer zoals bij SAVE en LOAD:

#### **K d:filenaam**

d: is weer de drivenaam, dit mag worden weggelaten. Ook hier is het gebruik van wildcards toegestaan.

## HET PROGRAMMA VERLATEN (E)

Door het intypen van de letter **E** wordt het programma verlaten.

Was dit niet de bedoeling of wordt het programma door een andere oorzaak verlaten, dan is het mogelijk om het programma met behoud van gegevens te starten door het commando

### GOTO 9880

te geven. Het is wel zaak niet eerst iets anders te doen na het verlaten van het programma.

### Opmerkingen:

Lege records worden automatisch gewist. Is het om de een of andere reden toch gewenst om lege records te hebben, dan kunnen deze gemaakt worden door in het eerste veld een spatie in te typen.

F1 is altijd terug naar het hoofdmenu.

Het is mogelijk de opdrachtregel te wijzigen. Hiertoe kunnen alle verandermogelijkheden van het TOEVOEGEN VAN GEGEVENS gebruikt worden.

F2 herhaalt de laatst ingevoerde opdracht.

Voor gebruik van het programma op een MSX-II dienen de volgende regels veranderd te worden:

```
125 CLEAR 4000:ON ERROR GOTO 9530
1340 MDAT = INT(800/BDAT + 2):MK = 3700
```

### Programma:

```
10 '*****
20 '**   D A T A B E S T A N D   **
30 '**   Versie 2.2.m2 (MSX-II)  **
40 '**   Door   J.Ribberink     **
100 '*****
110 '*** initialisatie
125 CLEAR 6400 : ON ERROR GOTO 9530
130 DEFINT A-J,L-X,Z : DIM TT(12,3),LS(9),TV$(12)
140 IA$=SPACE$(255) : MV=127 : VR=1
150 SM=15 : SW=40 : SCREEN 0 : WIDTH(SW)
160 C1$=CHR$(8)+CHR$(127)+CHR$(18)+CHR$(29)+CHR$(28)+CHR$(5)
170 C2$=CHR$(13)+CHR$(31)+CHR$(30)+CHR$(6)+CHR$(2)+CHR$(17)+CHR$(23)
180 C3$=CHR$(127)+CHR$(15)+CHR$(13)
190 C4$="IVTPMSLCKNRE-"
200 C5$="InsertVeranderTypePrintMenuSaveLoadCatalogKillNieuwR
    angshikExit-"
210 C6$=CHR$(32)+";:<>=( )-"+CHR$(13)
220 KEY 1,CHR$(17) : KEY 10,CHR$(23)
```



```

230 KEY 2,CHR$(15) : KEY 3,CHR$(5)+CHR$(13) : KEY 8,CHR$(5)
240 KEY 4,CHR$(2) : KEY 5,CHR$(6)
250 KEY 6,CHR$(19) : KEY OFF
260 KEY 7,"" : KEY 9,""
310 '*** Hoofdbesturing
330 GOSUB 630
340 INTERVAL OFF : LOCATE 0,23 : PRINT"Opricht :";
342 IF FO<>0 THEN BEEP : RESTORE930 : FOR J=1 TO FO : READ A$
: NEXT
343 IF FO>7 THEN A$=A$+STR$(FZ)+" in regel"+STR$(FL)
345 LOCATE 0,22 : IF FO<>0 THEN PRINT"Probleem : ";A$;
347 FO=0 : PRINT TAB(SW-1);
350 ZE=0 : MID$(IA$,1)=SPACE$(255) : IB$=""
360 CX=10 : CY=23 : ZP=1 : GOSUB 1560
370 GOSUB 1530 : CODE=INSTR(C3$,IB$)
380 ON CODE+1 GOTO 370,370,390,400
390 MID$(IA$,1)=OLD$ : ZE=LEN(OLD$) : GOTO 360
400 OLD$=LEFT$(IA$,ZE) : MID$(IA$,ZE+2)=CHR$(13)
410 I=1 : GOSUB 2830 : IF A$="" THEN 330 ELSE MID$(IA$,1)=MID
$(IA$,I)
420 GOSUB 1830 : MCODE=INSTR(C4$,CHR$(ASC(LEFT$(A$,1))-32))
: IF MCODE=0 THEN FO=1 : GOTO 340
430 I=INSTR(C5$,MID$(C4$,MCODE,1)) : J=INSTR(C5$,MID$(C4$,MCO
DE+1,1))
435 LOCATE , ,0
440 H$=MID$(C5$,I,J-I) : IF MID$(H$,2,LEN(A$)-1)<>MID$(A$,2)
THEN FO=1 : GOTO 340
450 IF MCODE<=4 THEN GOSUB 2030 : IF MCODE=1 THEN VER=0 : SR=
1 : ELSE VER=-1
455 WP=0 : FP=-1 : IF FO<>0 THEN 340
460 ON MCODE GOSUB 7040,7040,4040,4240,630,8030,8530,9030,913
0,1130,5030,9830
465 IF MCODE=11 THEN GOSUB 630
470 IF MCODE>4 THEN 340
475 PR=0 : SC=1 : SM=15 : IF MCODE=4 THEN SM=60 : PR=-1
480 INTERVAL ON
490 IF MCODE>1 THEN GOSUB 3040
500 IF WP<=FP THEN 500 ELSE IF MCODE>2 THEN 340
510 INTERVAL OFF : IF SO>0 THEN GOSUB 5070
520 GOSUB 3930 : GOTO 340
610 '*** Print menu
630 H$="DATABESTAND" : GOSUB 1030
640 IF SO=0 THEN A$="niet gesorteerd" : GOTO 670
650 IF LEFT$(DAT$(0,SO-1),1)=". " THEN A$="numeriek " ELSE A$=
""
660 A$=A$+"gesorteerd op "+MID$(DAT$(0,SO-1),2)
670 LOCATE 0,3 : PRINT A$ : LOCATE SW-10,3 : PRINT"Vrij :";MD
AT-VR+1
680 LOCATE 0,5 : PRINT "Menu : "
690 RESTORE 730 : FOR I=7 TO 18
700 READ A$ : LOCATE 3,I : PRINT A$
710 NEXT
720 RETURN
730 DATA "Nieuw : Maak een nieuw datablok"
740 DATA "Insert : Voeg records toe"
750 DATA "Verander : Verander veldgegevens"

```

```

760 DATA "Type      : Print op het scherm"
770 DATA "Print     : Print op de printer"
780 DATA "Rangschik: Sorteert het bestand"
790 DATA "Save      : Save op diskette"
800 DATA "Load      : Load van diskette"
810 DATA "Kill      : Wis een file van disk"
820 DATA "Catalog   : Druk alle files af"
830 DATA "Menu      : Druk dit menu af"
840 DATA "Exit      : Beeindig dit programma"
920 '*** Foutmeldingen
930 DATA syntax error,veldnaam is niet bekend,aantal haakjes
klopt niet
940 DATA de formule is te moeilijk,Database is vol
950 DATA Disk is vol,File is niet bekend
960 DATA Disk-error,error
1010 '*** Print header
1030 CLS : LOCATE 0,1
1040 A$=STRING$(SW/2,"*") : A$=A$+" "+H$+" "+A$
1050 PRINT MID$(A$,INT(LEN(A$)/2)-INT(SW/2)+1,SW)
1060 RETURN
1110 '*** Maak nieuw bestand
1130 H$="NIEUW BESTAND" : GOSUB 1030
1140 IF MK=0 THEN 1200
1150 LOCATE 0,22 : PRINT"Er staat al een bestand in het geheu
gen"
1160 BEEP : PRINT"Moet dit gewist worden ? (J/N) ";
1170 LOCATE ,,1 : A$=INPUT$(1) : LOCATE ,,0
1180 IF A$<>"j" AND A$<>"J" THEN RETURN ELSE GOSUB 1030
1200 LOCATE 0,3 : INPUT"Aantal velden ";BDAT
1210 IF (BDAT<=0) OR (BDAT>MV) THEN BEEP : GOTO 1200
1220 GOSUB 1330 : CK=0 : VR=1 : LOCATE 0,5
1230 FOR I=1 TO BDAT : DISP(I,0)=I : DAT$(0,I-1)=" Veld"+STR$
(I) : NEXT
1240 PA=BDAT-1 : FP=0 : WP=0 : VER=-1 : WERK(0)=0
1250 INTERVAL OFF : GOSUB 7040 : INTERVAL ON
1260 IF WP<=FP THEN 1260
1270 RETURN
1310 '*** Maak de dataarrays
1330 IF MK<>0 THEN ERASE DAT$,RANG,WERK,DISP
1340 MDAT=INT(800/(BDAT+2)) : MK=6000
1370 DIM DAT$(MDAT,BDAT-1),RANG(MDAT),WERK(MDAT),DISP(BDAT+2,
1)
1400 FOR I=1 TO MDAT : RANG(I)=I : NEXT
1410 RETURN
1510 '*** Maak inputstring
1530 IB$=INKEY$ : IF IB$="" THEN RETURN
1540 CODE=INSTR(C1$,IB$) : IF (ASC(IB$)<32 AND CODE=0) THEN R
ETURN
1550 ON CODE+1 GOSUB 1590,1630,1640,1650,1660,1670,1690
1560 I=ZP+CX-SW+3 : IF I<1 THEN I=1
1570 LOCATE CX,CY,0 : PRINT MID$(IA$,I,SW-CX-1);
1580 LOCATE CX+ZP-I,CY,1 : RETURN
1590 IF ZP>250 THEN RETURN
1600 IF INS THEN MID$(IA$,ZP)=IB$+MID$(IA$,ZP,250-ZP) : ZE=ZE
+1 ELSE MID$(IA$,ZP,1)=IB$
1610 IF ZE>250 THEN ZE=250 : ZP=ZE ELSE IF ZP>ZE THEN ZE=ZP
1620 ZP=ZP+1 : RETURN

```

```

1630 GOSUB 1660
1640 INS=-1 : MID$(IA$,ZP)=MID$(IA$,ZP+1) : ZE=ZE-1 : IF ZE<Z
P THEN ZE=ZP-1
1650 INS=NOT INS : RETURN
1660 ZP=ZP-2 : IF ZP<0 THEN ZP=0
1670 ZP=ZP+1 : IF ZP>ZE THEN ZP=ZE+1
1680 INS=0 : RETURN
1690 ZE=ZP-1 : MID$(IA$,ZP)=SPACE$(256-ZP) : RETURN
1810 '*** Maak a$ lowercase
1830 IF A$="" THEN RETURN
1840 FOR Z=1 TO LEN(A$)
1850 H=ASC(MID$(A$,Z,1)) : IF H<=90 AND H>=65 THEN MID$(A$,Z,
1)=CHR$(H OR 32)
1860 NEXT
1870 RETURN
2010 '*** Converteer opdrachtstring
2030 I=1 : TP=0 : LP=0 : PA=0 : BE=-1 : EI=VR-1 : FO=0 : REG=
1 : DISP(PA,1)=0
2040 GOSUB 2830 : IF K<>9 THEN 2070
2050 GOSUB 2930 : IF GTL THEN BE=VAL(A$) ELSE IF A$="" THEN B
E=1 ELSE FO=1 : RETURN
2060 I=I+1 : GOSUB 2830
2070 GOSUB 2930 : IF GTL THEN EI=VAL(A$) : GOSUB 2830 : IF BE
<0 THEN BE=EI
2080 IF EI>VR-1 THEN EI=VR-1
2090 IF BE>EI THEN BE=EI ELSE IF BE<0 THEN BE=1
2100 IF A$="" THEN 2150
2110 HA$=A$ : GOSUB 1830 : IF A$="als" THEN 2190
2120 IF A$="no" THEN DISP(PA,0)=-1 : PA=PA+1 : DISP(PA,1)=0 :
GOTO 2150
2130 GOSUB 2730 : IF A<0 THEN IF K<>10 THEN FO=2 : RETURN
2140 IF A<0 THEN TT(0,0)=SO-1 : TT(0,1)=3 : TT(0,2)=-1 : TV$(
0)=HA$ : GOTO 2360 ELSE DISP(PA,0)=A : PA=PA+1 : DISP(PA,1)=0
2150 IF K=2 THEN DISP(PA,1)=DISP(PA,1) OR 2
2160 IF K=3 THEN DISP(PA,1)=DISP(PA,1) OR 1
2170 IF K<>0 THEN I=I+1
2180 IF K=10 THEN TT(0,0)=-2 : GOTO 2360 ELSE GOSUB 2830 : GO
TO 2100
2190 GOSUB 2830 : IF K<>7 THEN 2220
2200 LP=LP+1 : LS(LP)=TP : TP=TP+1 : IF (LP>9 OR TP>12) THEN
FO=4 : RETURN
2210 TT(TP-1,0)=TP : TT(TP-1,1)=0 : I=I+1 : GOTO 2190
2220 GOSUB 1830 : GOSUB 2730 : IF A<0 THEN FO=2 : RETURN
2230 IF (K<3 OR K>6) THEN FO=1 : RETURN
2240 TT(TP,0)=A : TT(TP,1)=K-3 : I=I+1
2250 K=INT((INSTR("=<=>=<>",MID$(IA$,I-1,2))+1)/2)
2260 IF K<>0 THEN TT(TP,1)=K+3 : I=I+1
2270 IF LEFT$(DAT$(TT(TP,0),0),1)=". " THEN TT(TP,1)=TT(TP,1)+
6
2280 GOSUB 2830 : TV$(TP)=A$ : A=TP
2290 IF K<>8 THEN GOSUB 2830 : GOTO 2320
2300 TT(A,2)=-1 : A=LS(LP) : LP=LP-1 : IF LP<0 THEN FO=3 : RE
TURN
2310 I=I+1 : GOSUB 2830 : IF K=8 THEN 2300
2320 IF K=10 THEN TT(A,2)=-1 : IF LP<>0 THEN FO=3 : RETURN EL
SE 2360

```

```

2330 TP=TP+1 : IF TP>12 THEN FO=4 : RETURN ELSE TT(A,3)=TP
2340 GOSUB 1830 : IF A$="en" THEN TT(A,2)=1 : GOTO 2190
2350 IF A$="of" THEN TT(A,2)=0 : GOTO 2190 ELSE FO=1 : RETURN
2360 PA=PA-1 : IF PA<0 THEN PA=BDAT-1 : FOR J=0 TO PA : DISP(
J,0)=J : DISP(J,1)=0 : NEXT J
2370 IF TT(0,0)<0 THEN TT(0,0)=TT(0,0)+1
2380 REG=1 : FOR J=0 TO PA : REG=REG-((DISP(J,1) AND 2)=0) :
NEXT
2390 RETURN
2710 '*** Bepaal of a$ een veldnaam bevat
2730 A=-2 : IF A$="" THEN RETURN
2740 B$= A$ : FOR J=0 TO BDAT-1
2750 A$=MID$(DAT$(0,J),2,LEN(B$)) : GOSUB 1830 : IF A$=B$ THE
N A=J : J=BDAT
2760 NEXT
2770 IF A>-1 THEN NUM=(LEFT$(DAT$(0,A),1)=".") ELSE NUM=0
2780 A$=B$ : RETURN
2810 '*** Haal woord uit ia$
2830 K=INSTR(C6$+CHR$(34),MID$(IA$,I,1)):IF K=1 THEN I=I+1 :
GOTO 2830
2840 J=I : IF K=11 THEN J=I+1 : GOTO 2890
2850 IF K=0 THEN I=I+1 : K=INSTR(C6$,MID$(IA$,I,1)):GOTO2850
2860 A$=MID$(IA$,J,I-J)
2870 IF K=1 THEN I=I+1 : K=INSTR(C6$,MID$(IA$,I,1)):GOTO 2870
2880 RETURN
2890 I=I+1 : IF MID$(IA$,I,1)=CHR$(34) THEN K=1 : GOTO 2860
2895 IF MID$(IA$,I,1)=CHR$(13) THEN K=10 : GOTO 2860 ELSE 289
0
2910 '*** Test of a$ uit getallen bestaat
2930 GTL=-1 : IF A$="" THEN GTL=0 : RETURN
2940 FOR J=1 TO LEN(A$) : B=ASC(MID$(A$,J,1)) : GTL=(GTL AND
(B>47 AND B<58)) : NEXT
2950 RETURN
3010 '*** Zoek alles dat aan de voorwaarden voldoet
3040 FOR SR=BE TO EI
3050 LP=0 : TP=0 : TST=-1 : IF TT(0,0)=-1 THEN 3100
3060 IF TT(TP,1)=0 THEN LS(LP)=TP : TP=TT(TP,0) : LP=LP+1 : G
OTO 3060
3070 GOSUB 3230 : 'Test
3080 IF TT(TP,2)+TST=0 THEN TP=TT(TP,3) : GOTO 3060
3090 LP=LP-1 : IF LP>=0 THEN TP=LS(LP) : GOTO 3080
3100 IF TST=-1 THEN WERK(FP+1)=SR : FP=FP+1
3110 NEXT
3120 RETURN
3210 '*** Test een veld
3230 ON TT(TP,1) GOTO 3240,3250,3260,3270,3280,3290,3300,3310
,3320,3330,3340,3350
3240 TST=(DAT$(RANG(SR),TT(TP,0))<TV$(TP)) : RETURN
3250 TST=(DAT$(RANG(SR),TT(TP,0))>TV$(TP)) : RETURN
3260 TST=(DAT$(RANG(SR),TT(TP,0))=TV$(TP)) : RETURN
3270 TST=(DAT$(RANG(SR),TT(TP,0))<=TV$(TP)) : RETURN
3280 TST=(DAT$(RANG(SR),TT(TP,0))>=TV$(TP)) : RETURN
3290 TST=(DAT$(RANG(SR),TT(TP,0))<>TV$(TP)) : RETURN
3300 TST=(VAL(DAT$(RANG(SR),TT(TP,0)))<VAL(TV$(TP))) : RETURN
3310 TST=(VAL(DAT$(RANG(SR),TT(TP,0)))>VAL(TV$(TP))) : RETURN
3320 TST=(VAL(DAT$(RANG(SR),TT(TP,0)))=VAL(TV$(TP))) : RETURN

```

```

3330 TST=(VAL(DAT$(RANG(SR), TT(TP, 0))) <= VAL(TV$(TP))) : RETUR
N
3340 TST=(VAL(DAT$(RANG(SR), TT(TP, 0))) >= VAL(TV$(TP))) : RETUR
N
3350 TST=(VAL(DAT$(RANG(SR), TT(TP, 0))) <> VAL(TV$(TP))) : RETUR
N
3710 '*** Rangschik records na verander/insert
3740 RP=0 : IF SO=0 THEN 3820
3750 RP=RP+1 : IF RP>=VR THEN 3820
3770 RW=RP : IF DAT$(RANG(RW), SO-1)="" THEN 3750 ELSE 3790
3780 SWAP RANG(RW), RANG(RW+1)
3790 RW=RW-1 : IF RW<=0 THEN 3750
3800 IF DAT$(RANG(RW), SO-1)>DAT$(RANG(RW+1), SO-1) THEN 3780
3810 IF DAT$(RANG(RW), SO-1)="" THEN 3780 ELSE 3750
3820 RETURN
3910 '*** Zet lege records in lijst
3930 RW=VR-1
3940 I=RANG(RW)
3950 T=-1 : FOR J=0 TO BDAT-1 : T=(T AND (DAT$(I, J)="")) : NE
XT
3960 IF T THEN VR=VR-1 : IF VR<1 THEN VR=1 ELSE SWAP RANG(VR)
, RANG(RW)
3970 RW=RW-1 : IF RW>0 THEN 3940
3980 RETURN
4010 '*** Print een aantal records
4020 '*** Insprong vanuit timer
4040 ON INTERVAL=10 GOSUB 4050 : RETURN
4050 A$=INKEY$
4060 IF A$=CHR$(30) THEN WP=WP-14/(PA+1) : SC=1 : IF WP<0 THE
N WP=0
4070 IF A$=CHR$(31) THEN SC=1
4080 IF A$=CHR$(19) THEN SC=1 : WP=0
4090 IF A$=CHR$(15) THEN 4240
4100 IF A$=CHR$(17) THEN SR=VR : WP=FP+1 : RETURN
4110 IF (WP>FP) OR (SC>SM) THEN RETURN
4120 IF (SC+REG>SM) AND (REG<SM) THEN RETURN
4130 GOSUB 4330 : SC=SC+REG : WP=WP+1 : RETURN
4210 '*** Print aan een stuk door
4220 '*** Insprong vanuit timer
4240 ON INTERVAL=10 GOSUB 4250 : RETURN
4250 A$=INKEY$ : IF A$<>"" THEN 4040
4260 IF WP>FP THEN RETURN
4270 IF SC+REG>SM THEN SC=1
4280 GOSUB 4330 : WP=WP+1
4290 SC=SC+REG : RETURN
4310 '*** Druk af op scherm/printer
4330 IF SC=1 THEN GOSUB 1030 : LOCATE 0, 5
4340 OPEN "CRT:" FOR OUTPUT AS #1 : GOSUB 4430 : CLOSE #1
4350 IF PR THEN OPEN "LPT:" FOR OUTPUT AS #1 : GOSUB 4430 : C
LOSE #1
4360 RETURN
4410 '*** Druk een record af
4430 PRINT #1, ""
4440 FOR I=0 TO PA
4450 IF (DISP(I, 1) AND 2)=0 THEN PRINT #1, "" ELSE PRINT #1, "
"; : GOTO 4470
4460 IF I<>0 AND MCODE<>4 AND MCODE<>6 THEN PRINT #1, SPC(3);

```

```

4470 IF (DISP(I,1) AND 1)=0 THEN 4510
4480 IF DISP(I,0)=-1 THEN PRINT #1,"No"; : GOTO 4500
4490 PRINT #1,MID$(DAT$(0,DISP(I,0)),2);
4500 PRINT #1," ";
4510 IF DISP(I,0)=-1 THEN PRINT #1,WERK(WP); ELSE PRINT #1,DA
T$(RANG(WERK(WP)),DISP(I,0));
4520 NEXT I
4530 RETURN
5010 '*** Sorteert compleet bestand
5030 T=SO : I=1 : GOSUB 2830 : GOSUB 1830
5040 IF A$="op" THEN GOSUB 2830 : GOSUB 1830
5050 GOSUB 2930 : IF GTL THEN SO=VAL(A$) ELSE GOSUB 2730 : SO
=A+1
5060 IF SO<0 OR SO>BDAT THEN SO=T : FO=1-GTL : GOTO 5100
5065 IF SO=0 THEN GOSUB 630 : GOTO 5100
5070 LOCATE 0,22 : PRINT"Het bestand wordt gesorteerd op :";T
AB(SW-1);
5080 LOCATE 0,23 : PRINT "      "+MID$(DAT$(0,SO-1),2,SW-7);TA
B(SW-1);
5090 GOSUB 3740
5100 RETURN
7010 '*** Nieuw record verander/insert
7020 '*** Insprong vanuit timer
7040 ON INTERVAL=10 GOSUB 7050
7050 IF ((WP<=FP) OR VER) THEN 7080
7060 IF VR>MDAT THEN FO=5 : RETURN
7070 LDAT=VR : WERK(WP)=VR : VR=VR+1 : FP=WP
7080 IF WP>FP THEN LOCATE 0,22 : PRINT"Even geduld A.U.B." :
RETURN
7090 LOCATE 0,22 : PRINT TAB(39)
7100 WDAT=RANG(WERK(WP)) : SP=1 : PS=0
7110 SA=PA : IF SA>SM THEN SA=SM
7120 PS=0 : GOSUB 7230
7130 ON INTERVAL=1 GOSUB 7340
7140 RETURN
7210 '*** Nieuwe pagina/set ia$
7230 GOSUB 1030 : LOCATE 28,3 : PRINT"Pagina : "SP : LOCATE 0,
3 : PRINT"Record : "WERK(WP)
7240 WV=PS : LOCATE 0,5
7250 FOR I=PS TO PS+SA : PRINT MID$(DAT$(0,DISP(I,0)),2) " : "
;LEFT$(DAT$(WDAT,DISP(I,0)),SW-LEN(DAT$(0,DISP(I,0)))-3) : NEX
T
7260 CY=5+WV-PS : I=DISP(WV,0) : CX=LEN(DAT$(0,I))+2
7270 ZP=1 : ZE=LEN(DAT$(WDAT,I)) : MID$(IA$,1)=DAT$(WDAT,I)+S
PACE$(255-ZE)
7275 CK=CK-ZE
7280 LOCATE 0,CY,0 : PRINT MID$(DAT$(0,I),2) " ? "LEFT$(IA$,SW
-CX-1)
7290 LOCATE CX,CY,1 : RETURN
7310 '*** Besturing insert/verander
7320 '*** Insprong vanuit timer
7340 GOSUB 1530 : IF IB$="" THEN RETURN
7350 CODE=INSTR(C2$,IB$) : IF CODE=0 THEN RETURN
7360 I=DISP(WV,0) : DAT$(WDAT,I)=LEFT$(IA$,ZE) : CK=CK+ZE
7370 LOCATE CX-2,CY,0 : PRINT": "LEFT$(IA$,SW-CX-1) : IF CK>M
K THEN FO=5 : RETURN

```

```

7380 ON CODE GOTO 7390,7390,7430,7460,7470,7490,7500
7390 WV=WV+1 : IF WV>PA THEN 7460
7400 IF WV<=PS+SA THEN 7260
7410 PS=PS+SA+1 : SP=SP+1 : SA=PA-PS : IF SA>SM THEN SA=SM
7420 GOTO 7230
7430 WV=WV-1 : IF WV<0 THEN 7470
7440 IF WV>=PS THEN 7260
7450 PS=PS-SA-1 : SP=SP-1 : SA=SM : GOTO 7230
7460 WP=WP+2 : IF RANG(WP-2)=0 THEN GOSUB 7630
7470 WP=WP-1 : IF WP<0 THEN WP=0
7480 ON INTERVAL=1 GOSUB 7040 : RETURN
7490 SR=LDAT : WP=FP+1 : RETURN
7500 LOCATE 0,22 : PRINT"Record wissen (j/n) ?"
7510 A$=INPUT$(1) : IF A$="j" THEN 7520
7515 LOCATE 0,22 : PRINT SPACES$(30) : GOTO 7390
7520 CK=CK+LEN(DAT$(WDAT,DISP(WV,0)))
7530 FOR J=0 TO BDAT-1 : CK=CK-LEN(DAT$(WDAT,J)) : DAT$(WDAT,
J)=" " : NEXT
7540 GOTO 7460
7610 '*** Maak record 0 in orde
7630 FOR WV=0 TO BDAT-1
7640 IF INSTR(". "+CHR$(32),LEFT$(DAT$(0,WV),1))=0 THEN DAT$(0
,WV)=CHR$(32)+DAT$(0,WV)
7650 NEXT
7660 RETURN
8010 '*** Save bestand
8030 GOSUB 8730 : IF F$="" THEN FO=1 : RETURN
8031 IF LEFT$(IA$,1)="," THEN MID$(IA$,1,1)=" " : GOTO 8033
8032 IF LEFT$(IA$,1)<>CHR$(13) THEN FO=1 : RETURN
8033 GOSUB 2030 : IF FO<>0 THEN RETURN
8034 LOCATE 0,22,0 : PRINT"Even geduld A.U.B";TAB(SW-1);
8040 GOSUB 3040
8048 A$=LEFT$(F$,3) : GOSUB 1830 : IF A$<>"cas" THEN GOSUB 82
30
8049 OPEN F$ FOR OUTPUT AS #1
8050 LOCATE 0,22,0 : PRINT"Saving ";F$;TAB(39);
8055 J=0 : FOR I=0 TO PA : IF (((SO-1)=DISP(I,0)) AND ((DISP(
I,1) AND 2)=0)) THEN J=SO : NEXT
8060 PRINT #1,FP+2,REG-1,J;
8070 WP=FP+1 : WERK(WP)=0 : GOSUB 4440 : 'save veldnamen
8080 FOR WP=0 TO FP : GOSUB 4440 : NEXT : 'save records
8090 PRINT #1,"" : CLOSE 1
8100 RETURN
8210 '*** Backup voor save
8230 IF MID$(F$,INSTR(F$,"."),4)<>".dat" THEN RETURN
8240 ON ERROR GOTO 8300
8250 KILL LEFT$(F$,INSTR(F$,"."))+ "dab"
8260 NAME F$ AS LEFT$(F$,INSTR(F$,"."))+ "dab"
8270 ON ERROR GOTO 9530
8280 RETURN
8300 IF ERR=53 THEN RESUME NEXT
8310 ON ERROR GOTO 9530 : GOTO 9530
8510 '*** Laadt een bestand
8530 GOSUB 8730 : IF A$="" THEN FO=1 : RETURN
8540 OPEN F$ FOR INPUT AS #1
8550 LOCATE 0,22,0 : PRINT"Loading ";F$;TAB(SW);
8560 CK=0 : INPUT #1,VR,BDAT,SO

```

```

8570 GOSUB 1330
8580 FOR I=0 TO VR-1
8590 FOR J=0 TO BDAT-1 : LINE INPUT #1,DAT$(I,J) : CK=CK+LEN(
DAT$(I,J)) : NEXT J
8600 NEXT I
8610 CLOSE #1
8620 GOSUB 630 : RETURN
8710 '*** Bepaal filenaam
8730 I=1 : GOSUB 2830 : IF K=3 THEN K=0 : GOSUB 2850
8740 IF A$<>" " THEN F$ = A$ ELSE 8760
8745 MID$(IA$,1)=MID$(IA$,I)
8750 I=INSTR(A$,".") : IF INSTR(A$,".")=0 THEN F$=LEFT$(A$,8+
I)+".dat"
8760 RETURN
9010 '*** Catalog files
9030 GOSUB 1030
9040 GOSUB 8730 : IF A$="" THEN A$="*.dat"
9050 FILES A$
9060 RETURN
9110 '*** Wis een file
9130 FT$=F$ : GOSUB 1030 : GOSUB 8730
9140 IF A$="" THEN FO=1 : RETURN
9150 LOCATE 0,22 : PRINT"Moet ik ";F$;" wissen (j/n) ?";
9160 B$=INPUT$(1) : IF B$<>"j" THEN RETURN
9170 LOCATE 0,22 : PRINT"Ik wis nu ";F$;TAB(SW-1);
9180 KILL F$
9190 F$=FT$ : RETURN
9510 '*** Error trapping
9530 IF ERR=3 THEN RESUME 340
9540 FO=9 : FZ=ERR : FL=ERL
9550 IF FZ>=50 THEN FO=8
9560 IF FZ=66 THEN FO=6
9570 IF FZ=53 THEN FO=7
9580 RESUME 9590
9590 RETURN
9810 '*** Beeindig programma
9830 ON ERROR GOTO 0
9840 CLS : LOCATE 0,21 : PRINT"U kunt het programma weer"
9850 PRINT"opstarten met behoud van gegevens"
9860 PRINT"met : 'goto 9880'"
9870 END
9880 ON ERROR GOTO 9530
9890 GOTO 330

```

*Checksums:*

Gebruik het programma van paragraaf 4.6.

10:014B	20:E68D	30:4471	40:7D0C	100:8152
110:1DE4	125:26B9	130:365A	140:804F	150:59F6
160:BE95	170:4091	180:FAD9	190:A322	200:FECA
210:CB82				blok-checksum : ECD3
220:511F	230:2016	240:1B66	250:8B31	260:FFA3
310:1BAD	330:AC6F	340:F366	342:CE2E	343:0614
345:C88E	347:D193	350:35D2	360:ABC2	370:409A
380:DC54				blok-checksum : 252E



390:8265	400:829C	410:0809	420:AE6B	430:41A2	
435:6B03	440:37D6	450:D150	455:5A57	460:B0F4	
465:40C4	470:61DA	475:A0E9	480:EF0A	490:5388	
500:535E				blok-checksum :	52B7
510:ADD8	520:5721	610:BAFB	630:D530	640:B2CB	
650:57DF	660:6F79	670:0214	680:A0F5	690:16D9	
700:2F79	710:1B68	720:6F76	730:BF0A	740:10D7	
750:2126				blok-checksum :	B2CD
760:03F1	770:0119	780:49A0	790:58C8	800:D1EB	
810:28D3	820:3D1F	830:4E87	840:8BB0	920:2B1B	
930:D904	940:5BFF	950:E604	960:24DB	1010:EBA6	
1030:A575				blok-checksum :	9610
1040:A32E	1050:3E19	1060:CC76	1110:81CA	1130:1D03	
1140:0B2B	1150:ECCD	1160:E143	1170:55D6	1180:9985	
1200:7259	1210:798F	1220:E803	1230:5152	1240:D73C	
1250:6A44				blok-checksum :	D623
1260:DFA1	1270:CEF6	1310:0A0F	1330:0C79	1340:7EC5	
1370:5C50	1400:B601	1410:CDF6	1510:4462	1530:9FEC	
1540:915D	1550:2463	1560:2C2F	1570:DB83	1580:427A	
1590:8E95				blok-checksum :	9618
1600:F392	1610:F355	1620:9647	1630:8102	1640:283A	
1650:3A00	1660:D9D1	1670:3E13	1680:9D2A	1690:9266	
1810:8811	1830:67F9	1840:465A	1850:089E	1860:34C8	
1870:D4F6				blok-checksum :	F0A1
2010:314F	2030:5D8D	2040:7C14	2050:792F	2060:969D	
2070:0A85	2080:BA2B	2090:6B90	2100:D801	2110:EF90	
2120:22C6	2130:3EF2	2140:9E93	2150:A595	2160:B598	
2170:84BF				blok-checksum :	866F
2180:2965	2190:8812	2200:6FAC	2210:31B2	2220:4CD6	
2230:FA08	2240:8B1F	2250:E4BA	2260:936B	2270:306F	
2280:1923	2290:A25E	2300:B669	2310:D023	2320:71C0	
2330:033B				blok-checksum :	AA20
2340:9EF5	2350:D941	2360:DACB	2370:6FF1	2380:4646	
2390:D2F6	2710:9BFD	2730:3112	2740:13F1	2750:A17A	
2760:3508	2770:1725	2780:482A	2810:3FC4	2830:9205	
2840:C0A9				blok-checksum :	CF2D
2850:DA9A	2860:ECD0	2870:FAAE	2880:D776	2890:802B	
2895:2A4B	2910:748F	2930:D5A2	2940:DD6E	2950:D6F6	
3010:6FFD	3040:ACB2	3050:ECCE	3060:E59D	3070:4E5E	
3080:836A				blok-checksum :	EE46
3090:4A52	3100:2594	3110:3368	3120:CF76	3210:EBCE	
3230:4D5C	3240:2C55	3250:4C56	3260:EC55	3270:092E	
3280:4930	3290:092F	3300:4A06	3310:4AA6	3320:4B16	
3330:84A7				blok-checksum :	8D96
3340:85E7	3350:86B7	3710:E3E4	3740:E549	3750:CA60	
3770:69A3	3780:0659	3790:7178	3800:91A5	3810:3EC6	
3820:D676	3910:4E8D	3930:B0B4	3940:69DC	3950:DDCD	
3960:26C5				blok-checksum :	D8A1
3970:3F09	3980:DA76	4010:407B	4020:808C	4040:5D5F	
4050:6888	4060:F7F4	4070:4D1B	4080:B3AC	4090:4C4C	
4100:B5FD	4110:2585	4120:CCBD	4130:BF8B	4210:70E7	
4220:848C				blok-checksum :	39B6
4240:6160	4250:DC06	4260:5012	4270:5B41	4280:4DB4	
4290:05CB	4310:29E3	4330:D012	4340:7AA7	4350:E980	
4360:D576	4410:3944	4430:515B	4440:45AA	4450:1FA5	
4460:0351				blok-checksum :	7DD3

4470:7163	4480:4703	4490:BE23	4500:264C	4510:5836	
4520:6A59	4530:D5F6	5010:D38C	5030:C847	5040:7A57	
5050:8233	5060:FF93	5065:F14B	5070:1079	5080:AE7E	
5090:A912				blok-checksum :	ABFE
5100:D276	7010:F63E	7020:8C8C	7040:D88E	7050:C684	
7060:4D97	7070:7547	7080:044D	7090:083C	7100:9D0B	
7110:B83C	7120:E1EC	7130:871A	7140:D876	7210:B9EC	
7230:407E				blok-checksum :	A681
7240:26CA	7250:E01B	7260:428D	7270:50CC	7275:6718	
7280:9ADA	7290:DB45	7310:3AF9	7320:928C	7340:8B3C	
7350:349A	7360:97A8	7370:1051	7380:12FF	7390:01CF	
7400:32AE				blok-checksum :	BCB2
7410:22D3	7420:65C9	7430:60BD	7440:6706	7450:5640	
7460:B6E2	7470:4BCD	7480:093B	7490:43A4	7500:4582	
7510:23DF	7515:3ECD	7520:29EE	7530:D464	7540:6DD7	
7610:061A				blok-checksum :	20D8
7630:89DB	7640:7957	7650:3728	7660:DE76	8010:DAD2	
8030:A96A	8031:E664	8032:AFE7	8033:262C	8034:8B2E	
8040:C4F6	8048:FE16	8049:801A	8050:2669	8055:BB16	
8060:1247				blok-checksum :	5090
8070:044F	8080:9EEB	8090:CDFF	8100:D876	8210:6D41	
8230:7099	8240:E382	8250:961D	8260:9CE6	8270:E39B	
8280:DD76	8300:B344	8310:B3A9	8510:D33E	8530:A956	
8540:B58E				blok-checksum :	F7E1
8550:9D18	8560:16DC	8570:F8F0	8580:D7AA	8590:9D72	
8600:6E59	8610:B854	8620:5AB1	8710:B697	8730:3B9A	
8740:5145	8745:8D99	8750:5E96	8760:E176	9010:337A	
9030:D0E4				blok-checksum :	879E
9040:18B9	9050:B47B	9060:DC76	9110:7C23	9130:19A4	
9140:C762	9150:E691	9160:FB7A	9170:FE2F	9180:DD30	
9190:4863	9510:2EBA	9530:A3E0	9540:56D4	9550:2F1C	
9560:155C				blok-checksum :	FC64
9570:105E	9580:28FC	9590:E2F6	9810:9D5B	9830:BC3E	
9840:D06C	9850:209B	9860:9EB8	9870:1C24	9880:E3AC	
9890:C9BB				blok-checksum :	E564

### 7.3 MAIL/MERGE

Door middel van dit programma is het mogelijk om met standaardbrieven te werken. Standaardbrieven zijn brieven waarvan de inhoud ongewijzigd blijft, maar alleen bepaalde gegevens, zoals naam en adres, veranderen. Dit natuurlijk zonder dat we dat telkens handmatig moeten veranderen.

De gegevens uit het Databestandenprogramma worden gekoppeld aan een brief die met het tekstverwerkingsprogramma is gemaakt.

Het blijkt dus al dat we in elk geval twee dingen nodig hebben: een brief (tekst) en een bestand.

Het bestand hoeft niet aan speciale voorwaarden te voldoen. In de brief moeten we echter aangeven welke gegevens we uit de database in de brief willen hebben en op welke plaats. Dat is heel gemakkelijk. In de tekst gebruiken we de veldnamen die we in de database aan de velden hebben toegekend en wel omhuld door het control-d karakter (zie hiervoor ook de tekstverwerker). Bijvoorbeeld:

Geachte **D**Naam**D**,

Hoe gaat het in **D**plaats**D**.

Naam en plaats zijn dan veldnamen zoals we die in de database gebruiken. Het maakt niet uit of we hoofd- of kleine letters gebruiken. Spaties tussen de beide control-d karakters worden genegeerd.

We starten het Mail/Merge-programma. Allereerst wordt, evenals bij de tekstverwerker, gevraagd naar een aantal printergegevens. Zie voor de uitleg de tekst bij de tekstverwerker. Vervolgens moeten we device- en filenamen opgeven. Nu de disk-drive c.q. cassetterecorder niet onderbreken voor het programma klaar is met z'n activiteit.

Bij de cassetterecorder dienen we de remote-plug te gebruiken.

De printerkop moet de eerste keer goed bovenaan de eerste pagina staan. De volgende papiertransporten worden door het programma zelf verzorgd.

Als we een beperkt deel van ons databestand een brief willen sturen, bijvoorbeeld alleen alle mensen in Enschede, dan moeten we de selectie al maken bij het opslaan van het databestand vanuit de database. Zie aldaar voor de manieren waarop dat mogelijk is.

Het resultaat van ons voorbeeld zou kunnen zijn:

Geachte Heer Jansen,

Hoe gaat het in ZWOLLE.

Geachte Heer Klaassen,

Hoe gaat het in ARNHEM.

Geachte Heer Peters,

Hoe gaat het in ZWOLLE.

Programma:

```
1 '*****
2 '*      MSX MAIL-MERGE      *
3 '*      VERSIE 1 1986      *
4 '*      H GROOT LIPMAN     *
5 '*****
6 '
10 CLEAR 500,45000!:DEFINT A-Z:DEF FNH(X)=X\256+2
55:DEF FNL(X)=X MOD 256-256*(X MOD 256<0)
20 GOSUB 5000:'PRINTERINSTELLINGEN
30 GOSUB 22000:'MACHINECODE
40 GOSUB 1000:'TEKST INLEZEN
50 GOSUB 2000:'DATA OPENEN
60 CLS:LOCATE10,5:PRINT"PROGRAMMA RUNT"
70 GOSUB 4000:'VIND DATAVELDEN
100 FOR B=1 TO AR-1
110 FOR J=0 TO AV-1:LINE INPUT #1,R$(J):NEXT
120 GOSUB 6000:'INSERT RECORD
130 GOSUB 7000:'PRINT
140 GOSUB 8000:'WIS RECORD
150 NEXT:CLS:END
1000 'TEKST INLEZEN
1002 CLS:LOCATE 10,7:PRINT"MAIL-MERGE"
1004 LOCATE 5,10:INPUT"Device      ":V$
1010 LOCATE 5,13:INPUT"Tekstfile  ":T$
1030 POKE TS,211:POKE TT,197:EK=TT:NS=EK-1:BK=TS:
SL=NS-BK:GOSUB 9030:U=USR2(BK):POKE TS+1,32:POKE
TT,197
1040 A$=V$+": "+T$
1050 IF LEN(V$)>1 THEN IF (ASC(V$) OR 32)=113 AND
(ASC(MID$(V$,2)) OR 32)=100 THEN _BLOAD(A$) ELSE
BLOAD A$ ELSE BLOAD A$
1060 NS=USR4(0):RETURN
2000 'DATA OPENEN
2010 LOCATE 5,16:INPUT "Datafile  ":D$
2020 OPEN V$+": "+D$+".DAT" FOR INPUT AS #1
2030 INPUT #1,AR,AV,U
2040 DIM VN$(AV-1),R$(AV-1)
2050 FOR X=0 TO AV-1:LINE INPUT #1,A$:GOSUB 3000:
VN$(X)=A$:NEXT
2060 RETURN
3000 'FORMATTEER A$
3010 IF LEFT$(A$,1)=" "THEN A$=MID$(A$,2):GOTO 30
10
```

```

3020 IF RIGHT$(A$,1)="" THEN A$=LEFT$(A$,LEN(A$)-1):GOTO3020
3030 FOR D=1 TO LEN(A$):C=ASC(MID$(A$,D)):IF 64<C AND 91>C THEN A$=LEFT$(A$,D-1)+CHR$(C+32)+RIGHT$(A$,LEN(A$)-D)
3040 NEXT:RETURN
4000 'VIND DATAVELDEN
4010 DIM P(50,1):NR=0
4020 FOR X=TS TO NS
4030 IF PEEK(X)<>196 THEN NEXT:RETURN
4040 U=USR1(X):A$=""
4050 IF PEEK(X)=196 OR PEEK(X+1)=197 THEN U=USR1(X):GOTO 4070
4060 A$=A$+CHR$(PEEK(X)):U=USR1(X):GOTO4050
4070 GOSUB 3000
4080 NR=NR+1:P(NR,0)=X:P(NR,1)=AV
4090 FOR Y=0 TO AV-1:IF VN$(Y)=A$ THEN P(NR,1)=Y
4100 NEXT Y
4110 IF P(NR,1)=AV THEN PRINT A$+" is een foute veldnaam.":END
4120 NEXT:RETURN
5000 'PRINTERINSTELLINGEN
5010 CLS:KEY OFF:LOCATE 10,7:PRINT"PRINTER"
5020 LOCATE 5,10:INPUT"Gewenste Breedte : ";PB
5030 LOCATE 5,13:INPUT"Papierregels : ";LP
5040 LOCATE 5,16:INPUT"Tekst regels : ";RP
5050 IF PB<40 OR PB>250 OR LP<RP OR LP>100 OR RP<20 THENPRINT:PRINT"Waarden niet geaccepteerd.":PRINT"Druk een toets."A$=INPUT$(1):GOTO5000
5060 RETURN
6000 'INSERT RECORD
6010 FOR N=NR TO 1 STEP -1
6020 W$=R$(P(N,1)):L=LEN(W$):PK=P(N,0)
6030 FOR M=1 TO L:U=USR9(PK):POKE PK,ASC(RIGHT$(W$,M)):NEXT
6040 NEXT:POKE TT,197:RETURN
7000 REM PRINT
7010 GOSUB 9000:S1=1:S0=0
7020 GOSUB 7500:LPRINT R$:IF S0=0 OR MD>1 GOTO 7020
7030 RETURN
7500 IF S1 THEN SL=(LP-RP)\2:TL=RP:EL=LP-TL-SL:S1=0:MD=1:CL=SL:POKE 56931!,PB:POKE 56932!,0:US=BK+1
7510 R$=SPACE$(PB)
7520 Y=VARPTR(R$):POKE 56929!,PEEK(Y+1):POKE 56930!,PEEK(Y+2)

```

```

7530 ON MD GOTO 7600, 7700, 7800, 7900
7600 CL=CL-1: IF CL=0 THEN MD=2: CL=TL
7610 RETURN
7700 US=USR5(US)
7710 XZ=INSTR(R$, CHR$(&HC5)): IF XZ THEN R$=LEFT$(
R$, XZ-1)+SPACE$(PB-XZ+1): S0=1: MD=3
7715 XZ=INSTR(R$, CHR$(&HD0)): IF XZ THEN R$=LEFT$(
R$, XZ-1)+SPACE$(PB-XZ+1)
7720 XZ=INSTR(R$, CHR$(&HC2)): IF XZ THEN R$=LEFT$(
R$, XZ-1)+SPACE$(PB-XZ+1): MD=3
7725 IF US>NS THEN S0=1: MD=3
7730 CL=CL-1: IF CL=0 THEN MD=4: CL=EL
7740 RETURN
7800 CL=CL-1: IF CL=0 THEN MD=4: CL=EL
7810 RETURN
7900 CL=CL-1: IF CL=0 THEN MD=1: CL=SL
7910 RETURN
8000 'WIS RECORD
8010 FOR N=1 TO NR
8020 FOR M=1 TO LEN(R$(P(N, 1))): U=USR1(P(N, 0)): NE
XT: NEXT: RETURN
9000 'INIT ROUTINES
9020 BK=TS: NS=USR4(0): EK=TT: SL=NS-BK
9030 POKE56935!, FNL(BK): POKE56936!, FNH(BK): POKE 5
6933!, FNL(EK): POKE 56934!, FNH(EK): POKE 56925!, FNL
(SL): POKE 56926!, SL\256
9040 RETURN
22000 REM DUMP MACHINECODE
22005 SUM=0: RESTORE 22000: CLS: LOCATE 10, 10: PRINT"
EVEN GEDULD"
22010 READ A$: IF A$="s" OR A$="S" THEN IF SUM=29
THEN GOTO 22100 ELSE PRINT "DATA ERROR": STOP
22020 Y=VAL("&H"+A$): IF LEN(A$)=4 THEN X=Y ELSE P
OKE X, Y: X=X+1: SUM=SUM XOR Y
22030 GOTO 22010
22100 DEFUSR1=56857!: DEFUSR2=56757!: DEFUSR4=56677
!: DEFUSR5=56307!: DEFUSR9=56180!
22105 TS=&HAF C8: TT=&HDAA8
22110 POKE 56941!, FNL(TS): POKE 56942!, FNH(TS): POK
E 56939!, FNL(TT): POKE 56940!, FNH(TT)
22120 RETURN
22220 REM 1: DELETE 1 KARAKTER
22230 DATA DE19, 2A, 6B, DE, ED, 5B, F8, F7, B7, ED, 52, 20,
06, 1A, FE, C5, C8, 18, 0E, E5, C1, 2A, F8, F7, 23, ED, B0, 1A, F
E, C5, 20, 01, 1B, 3E, 20, 12, C9

```

```

22240 REM 2:DELETE MEERDERE KARAKTERS
22250 DATA DDB5,2A,F8,F7,ED,4B,5D,DE,09,23,E5,2A,
6B,DE,D1,D5,B7,ED,52,E5,C1,03,E1,E5,ED,5B,F8,F7,1
3,ED,B0,E1,2B,2B,ED,5B,F8,F7,ED,52,E5,C1,ED,5B,6B
,DE,1B,2A,6B,DE,36,20,78,B1,C8,ED,B8,C9
22255 DATA ED,B0,E1,2B,7E,FE,C5,C0,36,20,2A,6B,E2
,36,C5,C9
22280 REM 4:ZOEK TEKSTEINDE
22290 DATA DD65,2A,6B,DE,7E,FE,20,28,0B,FE,C5,20,
0A,36,20,ED,5B,6B,DE,12,2B,18,ED,22,F8,F7,C9
22300 REM 5:UITVULLEN
22310 DATA DBF3,2A,F8,F7,ED,5B,61,DE,3A,63,DE,47,
7E,FE,20,20,07,23,7E,FE,20,20,01,2B,7E,E5,D5,E1,7
7,E1,13,23,FE,D0,28,08,FE,C2,28,04,10,EE,18,04,22
,F8,F7,C9
22320 DATA E5,DD,E1,E5,3E,20,ED,4B,63,DE,2A,61,DE
,ED,B1,E1,20,EA,DD,BE,00,20,0B,1B,D5,E1,BE,28,13,
DD,22,F8,F7,C9,1B,D5,E1,BE,28,08,36,20,DD,2B,2B,B
E,20,F8,DD,22,F8,F7,2A,61,DE,3E,20,BE,20,03,23,18
,FA
22330 DATA BE,28,03,23,18,FA,BE,20,03,23,18,FA,E5
,C1,ED,5B,63,DE,2A,61,DE,19,2B,B7,ED,42,D8,09,E5,
BE,20,03,2B,18,FA,ED,5B,61,DE
22340 DATA 1A,FE,20,20,03,13,18,F8,1A,FE,20,28,03
,13,18,F8,E5,B7,ED,52,E1,38,E5,D5,E5,ED,52,E5,C1,
03,E1,E5,E5,D1,13,ED,B8,E1,23,D1,C1,E5,B7,ED,42,E
1,C8,C5,18,CE
22410 REM 9:INSERT 1 KARAKTER
22420 DATA DB74,2A,6B,DE,ED,4B,F8,F7,B7,ED,42,C8,
E5,C1,2A;6B,DE,ED,5B,6B,DE,2B,1A,FE,C5,20,06,7E,F
E,D3,C8,36,C5,ED,B8,3E,20,12,C9
22999 DATA S
30000 INPUTA$:PRINTA$,PEEK(VAL("&h"+A$)):GOTO30000
0

```

## 8 ASSEMBLER/DISASSEMBLER

Een groot voordeel van programma's geschreven in machinecode is dat deze veel sneller werken dan dezelfde programma's geschreven in BASIC. Programmeren in machinecode is echter veel lastiger en stelt hogere eisen aan de programmeur.

Het nu volgende programma is een handig stuk gereedschap voor degenen die zich thuisvoelen in de wereld van de machinecodes.

### INTYPEN EN STARTEN

In het programma wordt onderscheid gemaakt tussen hoofd- en kleine letters en deze moeten daarom ook precies zo worden ingetoetst. Met name geldt dit voor de DATA-regels 50000 - 52220. De REM-regels hoeven niet te worden ingetoetst maar zijn wel nuttig als men de werking van het programma wil begrijpen.

Een klein gedeelte van het programma is in machinecode geschreven. Deze machinecode bevindt zich in de DATA-regels 53000 en 53100. Deze regels worden door het programma gecontroleerd op typfouten. Indien op het beeldscherm de foutmelding:

**Data-error in regel 53000 of 53100**

verschijnt, dient men deze regels te controleren op typfouten.

Indien we gebruik maken van een cassetterecorder in plaats van een diskdrive, dan moet regel 28030 worden gewijzigd. De devicenaam „A:” moet worden vervangen door „cas:”. Deze wijziging is ook in de REM-regels aangegeven.

Nadat het programma zorgvuldig is ingetoetst en is opgeslagen op cassette of diskette, kunnen we het programma runnen. Na enige seconden verschijnt het volgende menu op het beeldscherm:



## MENU

```
a.....assembleren  
d.....disassembleren  
c.....copieren  
r.....runnen  
p.....poken  
z.....zoeken  
g.....geheugendump  
k.....karakterdump  
s.....saven  
l.....laden
```

>

```
end    menu    ->hex pr.uit run
```

Dit menu geeft een volledig overzicht van alle mogelijkheden die dit programma biedt. Onderaan het scherm zien we de zogenaamde prompt (>). Het programma wacht nu op een opdracht. We geven een opdracht door de betreffende letter in te toetsen, gevolgd door een aantal parameters. Voordat we iedere menu-keuze met bijbehorende parameters toelichten, eerst enige algemene informatie:

- Alle invoer dient in **kleine letters** te worden ingetoetst, hoofdletters worden genegeerd.
- Getallen mogen zowel in decimale, hexadecimale, octale of binaire notatie worden ingevoerd. Hexadecimale getallen moeten worden voorafgegaan door een dollarteken (\$) of &h, octale getallen moeten worden voorafgegaan door &o en binaire getallen door &b.
- Door middel van functietoets F3 kunnen we kiezen tussen decimale of hexadecimale weergave van getallen op beeldscherm of printer. Standaard worden getallen hexadecimaal weergegeven. Door eenmaal indrukken van F3 wordt gewisseld naar decimale notatie. Nogmaals indrukken van F3 geeft weer hexadecimale notatie.
- Op dezelfde wijze als we met F3 kunnen kiezen tussen decimale of hexadecimale notatie, kunnen we met functietoets F4 de printer aan of uit schakelen. Printer „aan” betekent dat alle uitvoer naar beeldscherm ook naar printer wordt gestuurd.
- Met functietoets F2 roepen we het menu op en met functietoets F1 beëindigen we een gekozen opdracht. F1 wordt ook gebruikt om het programma te beëindigen, maar dit werkt alleen als het programma wacht op een opdracht (dit is te zien aan de prompt „>”).
- Net als in MSX-BASIC mogen we de cursor vrij over het beeldscherm bewegen. Op deze wijze kunnen we een opdracht nogmaals uitvoeren door met de cursor terug te lopen en op <RETURN> te drukken. Ook kunnen we zo, tijdens het assembleren of disassembleren, met de cursor terug lopen, wijzigingen aanbrengen in de instructies en deze doorvoeren door vervolgens op <RETURN> te drukken. Beëindig de opdracht weer met functietoets F1.

In het nu volgende zullen we alle opdrachten nader toelichten.

## ASSEMBLEREN

Syntax: **a** beginadres

Met deze opdracht kunnen we een machinecode-programma in mnemonische code invoeren. Deze mnemonische code wordt meteen naar opcode vertaald en in het geheugen geplaatst. Het beginadres van dit geheugengebied geven we bij de aanroep van de opdracht mee.

Dit adres wordt in decimale of hexadecimale notatie, afhankelijk van de met F3 gekozen instelling, weergegeven op het beeldscherm. Daarachter voeren we de instructie in in het normale Z80-formaat, gevolgd door <RETURN>.

Bij foutieve invoer verschijnt de melding „error” en de instructie moet opnieuw worden ingevoerd.

Bij correcte invoer wordt de instructie naar opcode vertaald en in het geheugen geplaatst. De opcode wordt direct op het beeldscherm getoond. Het adres wordt vervolgens met het juiste aantal bytes verhoogd en weergegeven op het beeldscherm. Nu kan de volgende instructie worden ingevoerd.

We beëindigen de opdracht met de instructie „end” of de functietoets F1.

Het is toegestaan met de cursor terug te lopen en veranderingen aan te brengen, mits we iedere verandering beëindigen met <RETURN>. Beëindig de opdracht ook nu weer met F1.

*Voorbeeld:*

```
>a $c000
.$C000 ld hl,$f663      ' 21 63 F6
.$C003 ld a,1          ' 3E 01
.$C005 add a,2         ' C6 02
.$C007 ld c,a         ' 4F
.$C008 ld b,0         ' 06 00
.$C00A ld (hl),$2     ' 36 02
.$C00C ld ($f7f8),bc  ' ED 43 F8 F7
.$C010 end
```

*Opmerking:*

Het programma accepteert bij relatieve jumps zowel de relatieve waarde als het absolute adres. Een relatieve waarde moet worden voorafgegaan door een plus- of minteken en moet liggen tussen -128 en +127.

## DISASSEMBLEREN

Syntax: **d** beginadres

Deze opdracht doet precies het omgekeerde van de opdracht „Assembleren”. De opcode in het geheugen wordt vertaald naar mnemonische code en deze wordt op het scherm of op de printer afgedrukt.

Het disassembleren begint bij het opgeven beginadres. Beëindig de opdracht met de functietoets F1.

Ook nu weer is het toegestaan met de cursor terug te lopen en veranderingen aan te brengen. Beëindig met F1.

*Voorbeeld:*

```
>d$c000
.$C000 ld hl,$F663      ' 21 63 F6
.$C003 ld a,$1         ' 3E 01
.$C005 add a,$2        ' C6 02
.$C007 ld c,a          ' 4F
.$C008 ld b,$0         ' 06 00
.$C00A ld (hl),$2      ' 36 02
.$C00C ld ($F7F8),bc  ' ED 43 F8 F7
```

## KOPIEER GEHEUGENGEBIED (COPY)

Syntax: c beginadres,nieuwadres,lengte

Met behulp van deze opdracht kunnen we de inhoud van een geheugengebied verplaatsen. Zo kunnen we bijvoorbeeld ruimte vrijmaken in een machinetaalprogramma om nieuwe instructies tussen te voegen.

Bij de aanroep van de opdracht geven we achtereenvolgens het beginadres van het te verplaatsen geheugengebied, het adres van het nieuwe geheugengebied en het aantal te verplaatsen bytes.

## RUN MACHINETAALPROGRAMMA

Syntax: r startadres

De machinetaalroutine met het opgegeven startadres wordt gedraaid. Het resultaat wordt weergegeven.

*Voorbeeld:*

```
>r$c000
Start usr op adres $C000? (j/n)
Resultaat: 3
```

## GEHEUGENDUMP

Syntax: g beginadres

Met deze opdracht wordt de inhoud van het geheugen dat begint bij het opgegeven beginadres in decimale of hexadecimale notatie weergegeven op het beeldscherm of op de printer.

Beëindig de opdracht met F1.

*Voorbeeld:*

```
>g$c000
$c000 21 63 F6 3E 01 C6 02 4F 06 00
$c00A 36 02 ED 43 F8 F7 00 00 00 00
$c014 00 00 00 00 00 00 00 00 00 00
```

## POKE IN GEHEUGEN

Syntax:     p adres,getal1 [,getal2 ... ]  
          p adres,"string

Hiermee kunnen we rechtstreeks getallen of karakters op geheugenplaatsen „poken“.

*Voorbeeld:*

```
>p$c010, "test
```

## ZOEKROUTINE

Syntax:     z beginadres,eindadres,getal1 [,getal2 ... ]  
          z beginadres,eindadres,"string

Deze opdracht zoekt in het opgegeven geheugengebied naar het voorkomen van de meegegeven getallenreeks of string. Alle gevonden beginadressen van de getallenreeks of string worden afgedrukt.

*Voorbeeld:*

```
>z$c000,$c01F, "test
$c010
```

## PRINT KARAKTERS

Syntax: k beginadres

Vanaf het opgegeven beginadres wordt de inhoud van het geheugen in ASCII-tekenen afgedrukt. Geheugenplaatsen waarin zich geen ASCII-code bevindt worden door een punt weergegeven. Beëindig de opdracht met F1.

*Voorbeeld:*

```
>k$c000
$c000 !cv>.F.O..6.mCxwtest.....
```

## SAVE EN LOAD MACHINETAALPROGRAMMA

Syntax:       s filenaam,beginadres,eindadres  
              l filenaam

Een filenaam mag uit maximaal 6 tekens bestaan bij gebruik van een cassetterecorder. Bij gebruik van een diskdrive uit maximaal 8 tekens eventueel gevolgd door een punt en 3 extra tekens.

## GEHEUGENGEBRUIK

Aangezien dit programma direct in het geheugen werkt is het nuttig te weten hoe de geheugenindeling is. Het BASIC-hoofdprogramma neemt het geheugengedeelte tot &HBEEE (49151) beslag.

Vrij te gebruiken is het geheugengedeelte van &HC000 (49152) tot &HDC61 (56417) bij gebruik van een diskdrive of tot &HF37F (62335) bij gebruik van een cassetterecorder.

*Programma:*

```
1 REM | ASSEMBLER-DISASSEMBLER |
2 REM | DOOR J.GROOT LIPMAN |
3 REM | 1985 |
10 CLEAR 300,&HBDFF: DEFSNG A-Z
20 GOSUB 40000: GOSUB 100
30 GOTO 10000
99 'DEFFN's
100 DEFFNH$(X)=" "+STRING$(-(X<16),"0")+HEX$(X)
110 DEFFNLO(X)=X-256*INT(X/256)
120 DEFFNHI(X)=INT((X+65536!)/256) MOD 256
130 DEFFNI$(A$,X,I$)=LEFT$(A$,X)+I$+MID$(A$,X+1)
140 DEFFNDP(X)=PEEK(X)+256*PEEK(X+1)
170 DEFFNG$(X)="$"+HEX$(X): HEX=TRUE: GOTO 190
180 DEFFNG$(X)=MID$(STR$(X-65536!*(X<0)),2): HEX=FALSE
190 RETURN
199 'Put byte
200 POKE PU,BYTE: PU=PU+1
210 IF PU=65536! THEN PU=0
220 RETURN
299 'Get byte
300 BYTE=PEEK(GE): GE=GE+1
310 IF GE=65536! THEN GE=0
320 RETURN
399 'Get adres
400 L=INSTR(CO$,""): IF L=0 THEN L=LEN(CO$)+1
410 NR$=LEFT$(CO$,L-1): CO$=MID$(CO$,L+1)
420 GOSUB 800
430 ADRES=NR: RETURN
499 'Get bytes
500 AB=0
510 IF CO$="" THEN RETURN
520 IF LEFT$(CO$,1)=CHR$(34) THEN GOTO 570: 'String
```

```

530 L=INSTR(CO$,""): IF L=0 THEN L=LEN(CO$)+1
540 NR$=LEFT$(CO$,L-1): CO$=MID$(CO$,L+1)
550 GOSUB 800
560 BYTE=NR: GOSUB 200: AB=AB+1: GOTO 510
570 CO$=MID$(CO$,2): IF CO$="" THEN RETURN
580 BYTE=ASC(CO$): GOSUB 200: AB=AB+1
590 GOTO 570
599 'Poke dubbel
600 POKE DP, FNLO(ADRES): DP=DP+1
610 POKE DP, FNHI(ADRES): DP=DP+1
620 RETURN
699 'Get filename
700 L=INSTR(CO$,""): IF L=0 THEN L=LEN(CO$)+1
710 FI$=LEFT$(CO$,L-1): CO$=MID$(CO$,L+1)
720 RETURN
799 'Val(nr$)
800 SG$=LEFT$(NR$,1)
810 IF SG$="+" OR SG$="-" THEN NR$=MID$(NR$,2) ELSE SG$=""
820 P=INSTR(NR$,"$"): IF P THEN NR$=LEFT$(NR$,P-1)+"&h"+MID$(
NR$,P+1)
830 NR=VAL(NR$): IF SG$="-" THEN NR=-NR
840 IF NR<>0 THEN GT=TRUE: RETURN
850 IF RIGHT$(NR$,1)="0" THEN GT=TRUE: RETURN
860 GT=FALSE: RETURN
899 'Dec<->hex
900 IF CO$="->hex" THEN KEY 3, "->dec"+CHR$(13): GOTO 180
910 KEY 3, "->hex"+CHR$(13): GOTO 170
949 'Pr.aan<->uit
950 IF CO$="pr.aan" THEN KEY 4, "pr.uit"+CHR$(13): PR=FALSE: R
ETURN
960 KEY 4, "pr.aan"+CHR$(13): PR=TRUE: RETURN
999 'Assembler, subr, zoek in tab
1000 FO=FALSE
1010 FOR NR=0 TO 7
1020 IF T$(T,NR)<>Z$ THEN NEXT NR: RETURN
1030 IF T<8 THEN OC=T(T) OR 8*NR
1040 FO=TRUE: RETURN
1099 'Get reg
1100 FO=FALSE
1110 FOR REG=0 TO 7
1120 IF T$(11,REG)=R$ THEN FO=TRUE: RETURN
1130 NEXT REG: RETURN
1199 'Dispatch
1200 AB=1: IF SG$<>"" THEN B1=BYTE: GOTO 1220
1210 B1=BYTE-PU-2
1220 IF B1<-128 OR B1>127 THEN ERROR 253
1230 RETURN
1299 '1 Byte
1300 AB=1: B1=BYTE: RETURN
1399 '2 Byte
1400 AB=2
1410 B1=FNLO(BYTE): B2=FNHI(BYTE)
1420 RETURN
1499 'Index-reg
1500 O$="": P=INSTR(A$,"ix"): IF P THEN I$="x": GOTO 1530
1510 P=INSTR(A$,"iy"): IF P THEN I$="y": GOTO 1530

```

```

1520 I$="": RETURN
1530 MID$(A$,P,2)="hl": O$=MID$(A$,P+2,1)
1540 IF O$<>"+" AND O$<>"-" THEN O$="": GOTO 1560
1550 E=INSTR(P+3,A$,")": O$=MID$(A$,P+2,E-P-2):A$=LEFT$(A$,P
+1)+MID$(A$,E)
1560 RETURN
1599 'Opsplitsen
1600 P1=INSTR(A$, " "): P2=INSTR(A$,",",")
1610 IF P1=0 THEN P1=LEN(A$)+1
1620 IF P2=0 THEN P2=LEN(A$)+2
1630 IF P2<P1 THEN ERROR 251
1640 MN$=LEFT$(A$,P1-1)
1650 OP$=MID$(A$,P1+1)
1660 O1$=MID$(A$,P1+1,P2-P1-1)
1670 O2$=MID$(A$,P2+1)
1680 RETURN
1699 'Getal -> V
1700 NR$=O1$: GOSUB 1720: O1$=NR$: IF GT THEN RETURN
1710 NR$=O2$: GOSUB 1720: O2$=NR$: RETURN
1720 IF LEFT$(NR$,1)="(" THEN NR$=MID$(NR$,2,LEN(NR$)-2): GOS
UB 1730: NR$="(+NR$)": RETURN
1730 GOSUB 800
1740 IF GT THEN BYTE=NR: NR$="V"
1750 RETURN
1799 'Errors
1800 IF ERR<200 THEN ON ERROR GOTO 0 ELSE RESUME 1810
1810 BEEP: PRINT: PRINT "error": ER=TRUE
1820 RETURN
1999 'Assembler decoder
2000 ON ERROR GOTO 1800: GOSUB 1500: GOSUB 1600: IF ER THEN R
ETURN
2010 AB=0: ED$="": CB$=""
2099 'CB-codes,binair
2100 CB$="cb":T=14: Z$=O1$: GOSUB 1000: IF NOT FO OR O2$="" T
HEN 2200
2110 R$=O2$: GOSUB 1100: IF NOT FO THEN ERROR 255
2120 IF MN$="bit" THEN OC=64 ELSE IF MN$="res" THEN OC=128 EL
SE IF MN$="set" THEN OC=192 ELSE ERROR 255
2130 OC=OC OR 8*NR OR REG: RETURN
2199 'CB-aritmetiek
2200 T=5: Z$=MN$: GOSUB 1000: IF NOT FO THEN CB$="": GOTO 230
0
2210 R$=O1$: GOSUB 1100: IF NOT FO THEN ERROR 255
2220 OC=OC OR REG: RETURN
2299 'ED-codes,divers
2300 ED$="ed": T=13: Z$=A$: GOSUB 1000: IF FO THEN OC=ED(NR):
RETURN
2310 GOSUB 1700: 'Nu getal->V
2320 OP$=O1$: A$=MN$: IF O1$<>" THEN A$=A$+" "+O1$: IF O2$<>
"" THEN A$=A$+" "+O2$: OP$=OP$+" "+O2$
2399 'ED-ld
2400 IF MN$<>"ld" THEN 2500
2410 T=6: Z$=OP$: GOSUB 1000: IF FO THEN GOTO 1400
2420 T=7: Z$=OP$: GOSUB 1000: IF FO THEN RETURN
2499 'ED-aritmetiek
2500 IF MN$="sbc" THEN OC=66 ELSE IF MN$="adc" THEN OC=74 ELS
E 2600

```

```

2510 IF O1$<>"hl" THEN 2600
2520 T=10: Z$=O2$: GOSUB 1000: IF NOT FO THEN ERROR 255
2530 OC=OC OR 16*NR: RETURN
2599 'Ldi enz
2600 IF O1$<>" " OR MN$="cpl" THEN GOTO 2700
2610 IF LEFT$(MN$,3)="out" THEN MN$="ot"+MID$(MN$,4)
2620 T=12: Z$=LEFT$(MN$,2): GOSUB 1000: IF NR>3 THEN 2700
2630 N=NR: Z$=MID$(MN$,3): GOSUB 1000: IF NR<4 OR NOT FO THEN
  ERROR 255
2640 OC=160 OR (NR-4)*8 OR N: RETURN
2699 'Ed i/o
2700 IF O1$<>"(c)" OR MN$<>"out" THEN 2720
2710 OC=65: R$=O2$: GOTO 2740
2720 IF O2$<>"(c)" OR MN$<>"in" THEN 2760
2730 OC=64: R$=O1$
2740 GOSUB 1100: IF NOT FO THEN ERROR 255
2750 OC=OC OR REG*8: RETURN
2760 ED$="": 'Geen ed
2799 'Add hl,
2800 IF MN$<>"add" OR O1$<>"hl" THEN 2900
2810 T=10: Z$=O2$: GOSUB 1000: IF NOT FO THEN ERROR 255
2820 OC=9 OR 16*NR: RETURN
2899 'In T$(0-4)
2900 T=0: Z$=A$: GOSUB 1000: IF FO THEN IF NR=0 THEN 1400 ELSE
  E IF NR=2 OR NR=3 THEN 1300 ELSE RETURN
2910 T=1: Z$=A$: GOSUB 1000: IF NOT FO THEN 2920
2913 IF NR=2 OR NR=3 THEN 1200
2916 IF NR>3 THEN OC=201 OR 16*(NR-4)
2918 RETURN
2920 T=2: IF MN$="ld" THEN Z$=OP$: GOSUB 1000: IF NR<4 THEN R
  ETURN ELSE IF NR<8 THEN 1400
2930 T=3: Z$=MN$: GOSUB 1000: IF FO THEN RETURN
2940 T=4: Z$=MN$: GOSUB 1000: IF NOT FO THEN 3000
2950 IF O2$="" THEN R$=O1$ ELSE R$=O2$
2960 GOSUB 1100: IF NOT FO THEN OC=198 OR 8*NR: GOTO 1300
2970 OC=OC OR REG: RETURN
2999 'O1$ is 8 bit reg
3000 R$=O1$: GOSUB 1100: IF NOT FO THEN 3100
3010 IF MN$="inc" THEN OC=4 OR 8*REG: RETURN
3020 IF MN$="dec" THEN OC=5 OR 8*REG: RETURN
3030 IF MN$<>"ld" THEN IF O1$="c" THEN 3100 ELSE ERROR 255
3040 R=REG: R$=O2$: GOSUB 1100: IF FO THEN OC=64 OR 8*R OR RE
  G: RETURN
3050 IF O2$="V" THEN OC=6 OR 8*R: GOTO 1300
3060 ERROR 255
3099 'Conditities
3100 T=9: Z$=O1$: GOSUB 1000: IF NOT FO THEN 3200
3110 IF MN$="jr" AND NR<4 THEN OC=32 OR 8*NR: GOTO 1200
3120 IF MN$="jp" THEN OC=194 OR 8*NR: GOTO 1400
3130 IF MN$="call" THEN OC=196 OR 8*NR: GOTO 1400
3140 IF MN$="ret" THEN OC=192 OR 8*NR: RETURN
3150 ERROR 255
3199 'Rst
3200 IF MN$<>"rst" THEN 3300
3210 NR=INT(BYTE/8): OC=199 OR 8*NR: RETURN
3299 'Ld dubbel
3300 IF MN$<>"ld" THEN 3400

```



```

3310 T=10: Z$=O1$: GOSUB 1000: IF NOT FO THEN ERROR 255
3320 OC=1 OR 16*NR: GOTO 1400
3399 'inc 16reg
3400 IF MN$="dec" THEN OC=11 ELSE IF MN$="inc" THEN OC=3 ELSE
3500
3410 T=10: Z$=O1$: GOSUB 1000: IF NOT FO THEN ERROR 255
3420 OC=OC OR 16*NR: RETURN
3499 'Stack
3500 IF MN$="pop" THEN OC=193 ELSE IF MN$="push" THEN OC=197
ELSE 3600
3510 T=8: Z$=O1$: GOSUB 1000: IF NOT FO THEN ERROR 255
3520 OC=OC OR 16*NR: RETURN
3599 'Halt
3600 IF MN$="halt" THEN OC=118: RETURN
3699 'Call
3700 IF MN$="call" THEN OC=&HCD: GOTO 1400
3799 'Niet goed
3800 ERROR 254
4999 'Disassembler, subr, get byte
5000 BYTE=A(NR): NR=NR+1
5010 RETURN
5099 'Rst
5100 AD=((OC/8) AND 7)*8
5110 RE$=FNG$(AD): RETURN
5199 '1-byte
5200 GOSUB 5000
5210 RE$=FNG$(BYTE): RETURN
5299 '2-byte
5300 GOSUB 5000: AD=BYTE
5310 GOSUB 5000: AD=AD+256*BYTE
5320 RE$=FNG$(AD): RETURN
5399 'Jr-ofset
5400 GOSUB 5000: D=BYTE
5410 IF D>127 THEN D=D-256
5420 RE$=FNG$(GE+D+2): RETURN
5999 'Disassembler-decoder
6000 'DIM A(4)
6010 H=GE: FOR I=1 TO 4: GOSUB 300
6020 A(I)=BYTE: NEXT I: NR=1: GE=H
6030 GOSUB 5000: OC=BYTE
6040 IF BYTE=&HDD THEN I$="x": GOTO 6100
6050 IF BYTE=&HFD THEN I$="y": GOTO 6100
6060 I$="": GOTO 6200
6099 'index-reg
6100 OF=-1: GOSUB 5000: OC=BYTE
6110 IF OC=&H39 OR OC<&H34 OR OC>&HCC THEN 6200
6120 GOSUB 5000: OF=BYTE
6130 A(3)=A(4): 'Verwyder offset
6200 TI=0
6210 IF OC=&HCB THEN TI=25: GOSUB 5000: OC=BYTE: GOTO 6300
6220 IF OC=&HED THEN TI=30: GOSUB 5000: OC=BYTE
6300 FOR I=TI TO 39
6310 IF ((OC-MA(I,0)) AND MA(I,1)) <> 0 THEN NEXT I
6320 MA=MA(I,1): MN$=MN$(I)
6330 P=0
6399 'Vervang

```

```

6400 FOR S=LEN(MN$) TO 1 STEP -1
6410 P$=MID$(MN$,S,1)
6420 IF P$<"G" OR P$>"Z" THEN NEXT S: GOTO 6460
6430 GOSUB 7000
6440 MN$=LEFT$(MN$,S-1)+RE$+MID$(MN$,S+1)
6450 GOTO 6400
6460 IF P<>12 THEN 6500
6470 MN$=RIGHT$(MN$,2)+LEFT$(MN$,LEN(MN$)-2)
6480 IF MN$="otd" THEN MN$="outd"
6490 IF MN$="oti" THEN MN$="outi"
6499 'Index
6500 IF I$="" THEN 6600
6510 P=INSTR(MN$,"hl"): IF P=0 THEN 6600
6520 MID$(MN$,P,2)="i"+I$
6530 IF OF<0 THEN 6500
6540 MN$=FNI$(MN$,P+1,""+FNG$(OF))
6600 RETURN
6999 'Vervang
7000 IF P$>="V" THEN 7100
7005 IF P$="T" THEN 7200
7010 P=ASC(P$)-ASC("G")
7020 IF MA/2=MA\2 THEN 7040
7030 MA=MA\2: OC=OC\2: GOTO 7020
7040 I=NOT MA AND OC AND 7
7050 MA=MA OR 7: OC=OC OR 7
7060 RE$=T$(P,I)
7070 IF RE$="add" OR RE$="adc" OR RE$="sbc" THEN RE$=RE$+" a,
": MN$=MID$(MN$,2)
7080 RETURN
7099 'Bytes
7100 ON ASC(P$)-ASC("W")+1 GOTO 5100,5200,5300,5400
7109 'P$="V"
7110 A$=LEFT$(MN$,2)
7120 IF A$="in" OR A$="ou" THEN 5200
7130 IF A$="jr" OR A$="dj" THEN 5400
7140 GOTO 5300
7199 'Overig ed
7200 FOR I=0 TO 7
7210 IF OC=ED(I) THEN RE$=T$(13,I): RETURN
7220 NEXT I: RE$="??"
7240 RETURN
9999 'Main
10000 SCREEN 0: WIDTH 38
10010 PRINT TAB(15);"MENU"
10020 PRINT: PRINT: PRINT: PRINT
10030 PRINT TAB(7);"a.....assembleren"
10040 PRINT TAB(7);"d.....disassembleren"
10050 PRINT TAB(7);"c.....copieren"
10060 PRINT TAB(7);"r.....runnen"
10070 PRINT TAB(7);"p.....poken"
10080 PRINT TAB(7);"z.....zoeken"
10090 PRINT TAB(7);"g.....geheugendump"
10100 PRINT TAB(7);"k.....karakterdump"
10110 PRINT TAB(7);"s.....saven"
10120 PRINT TAB(7);"l.....laden"
10999 'Input command
11000 F(1)=FALSE: KEY (1) OFF

```

```

11005 IF PR THEN IF LPOS(0)<>0 THEN LPRINT: LPRINT
11010 PRINT: PRINT ">";: LINE INPUT CO$
11015 IF CO$="" THEN GOTO 11010
11020 KEY (1) ON
11030 IF CO$="end" THEN END
11040 IF CO$="menu" THEN GOTO 10000
11050 IF CO$="->hex" OR CO$="->dec" THEN GOSUB 900: GOTO 1100
0
11055 IF CO$="pr.aan" OR CO$="pr.uit" THEN GOSUB 950: GOTO 11
000
11060 C$=LEFT$(CO$,1): CO$=MID$(CO$,2)
11065 IF LEFT$(CO$,1)=" " THEN CO$=MID$(CO$,2):GOTO 11065
11070 IF C$=">" THEN GOTO 11030
11080 ON INSTR("adcrpzgksl.",C$)+1 GOSUB 11090,20000,21000,22
000,23000,24000,25000,26000,27000,28000,29000,30000: GOTO 110
00
11090 PRINT "Onbekend commando": BEEP
11100 RETURN
19999 'Assembler-main
20000 KEY (1) OFF: GOSUB 400: PU=ADRES
20010 PRINT ". ";FNG$(PU);TAB(7)
20020 LINE INPUT A$
20023 IF INSTR(A$,"end") THEN RETURN
20027 LOCATE 24,CSRLIN-1: PRINT "'";
20030 IF LEFT$(A$,1)<>"." THEN GOTO 20060
20040 NR$=MID$(A$,2,5): GOSUB 800
20050 PU=NR: A$=MID$(A$,8)
20060 P=INSTR(A$,"'"): IF P THEN A$=LEFT$(A$,P-1)
20070 IF RIGHT$(A$,1)=" " THEN A$=LEFT$(A$,LEN(A$)-1): GOTO 2
0070
20075 IF PR THEN LPRINT ". ";FNG$(PU);TAB(7);A$;TAB(24);"'";
20080 ER=FALSE: GOSUB 2000: IF ER THEN GOTO 20010 ELSE ON ERR
OR GOTO 0
20090 IF I$="x" THEN BYTE=&HDD: GOSUB 20200
20100 IF I$="y" THEN BYTE=&HFD: GOSUB 20200
20110 IF ED$="ed" THEN BYTE=&HED: GOSUB 20200
20120 IF CB$="cb" THEN BYTE=&HCB: GOSUB 20200
20130 B=LEN(ED$+CB$)/2: NR$=0$: GOSUB 800
20140 IF B AND O$<>"" THEN BYTE=NR: GOSUB 20200
20150 BYTE=OC: GOSUB 20200
20160 IF B=0 THEN IF O$<>"" THEN BYTE=NR: GOSUB 20200
20170 IF AB THEN BYTE=B1: GOSUB 20200
20180 IF AB=2 THEN BYTE=B2: GOSUB 20200
20190 PRINT: IF PR THEN LPRINT
20195 GOTO 20010
20199 'Put byte
20200 IF BYTE<0 THEN BYTE=BYTE+256
20210 PRINT FNH$(BYTE);: IF PR THEN LPRINT FNH$(BYTE);
20220 GOSUB 200: RETURN
20999 'Disassembler-main
21000 GOSUB 400: GE=ADRES: F(1)=FALSE
21020 GOSUB 6000
21030 PRINT ". ";FNG$(GE);TAB(7);MN$;TAB(24);"'";
21040 IF PR THEN LPRINT ". ";FNG$(GE);TAB(7);MN$;TAB(24);"'";
21050 FOR I=2 TO NR: 'Nr=aantal bytes+1
21060 GOSUB 300: PRINT FNH$(BYTE);

```

```

21070 IF PR THEN LPRINT FNH$(BYTE);
21080 NEXT I: PRINT: IF PR THEN LPRINT
21090 IF F(1) THEN RETURN
21100 GOTO 21020
21999 'Copy-main
22000 DP=&HBE30: GOSUB 400: GOSUB 600
22010 GOSUB 400: GOSUB 600
22020 GOSUB 400: GOSUB 600
22030 A=USR1(0): RETURN
22999 'Run-main
23000 GOSUB 400: DEFUSR=ADRES
23010 BEEP: PRINT "Start usr op adres ";FNG$(ADRES) ;"? (j/n)
"
23020 A$=INPUT$(1): IF A$<>"j" THEN RETURN
23030 PRINT "Resultaat: ";USR(0)
23040 RETURN
23999 'Poke-main
24000 GOSUB 400: PU=ADRES
24010 GOSUB 500
24020 RETURN
24999 'Zoek-main
25000 DP=&HBE80: GOSUB 400: GOSUB 600
25010 BEG=ADRES: GOSUB 400
25020 ADRES=ADRES-BEG+1: GOSUB 600
25030 PU=&HBF00: GOSUB 500: POKE &HBE86,AB
25040 IF F(1) OR NOT USR2(0) THEN RETURN
25050 A$=FNG$(FNDP(&HBE84))
25060 IF POS(0)>32 THEN PRINT
25070 IF PR THEN IF LPOS(0)>32 THEN LPRINT
25080 PRINT USING "\ \ ";A$;
25090 IF PR THEN LPRINT USING "\ \ ";A$;
25100 GOTO 25040
25999 'Geheugendump
26000 GOSUB 400: GE=ADRES
26010 IF PR THEN LPRINT FNG$(GE);TAB(6);
26020 PRINT FNG$(GE);TAB(6);
26030 IF HEX THEN GOTO 26100
26040 FOR X=1 TO 8: GOSUB 300
26050 IF PR THEN LPRINT USING "####";BYTE;
26060 PRINT USING "####";BYTE;: NEXT X
26070 IF F(1) THEN RETURN
26080 IF PR THEN LPRINT
26090 GOTO 26010
26100 FOR X=1 TO 10: GOSUB 300
26110 IF PR THEN LPRINT FNH$(BYTE);
26120 PRINT FNH$(BYTE);: NEXT X
26130 IF F(1) THEN RETURN
26140 PRINT: IF PR THEN LPRINT
26150 GOTO 26010
26999 'Karakterdump
27000 GOSUB 400: GE=ADRES
27010 IF PR THEN LPRINT FNG$(GE);TAB(6);
27020 PRINT FNG$(GE);TAB(6);
27030 FOR X=1 TO 32: GOSUB 300
27040 Y=BYTE MOD 128: IF Y<32 OR Y=127 THEN Y=ASC(".")
27050 PRINT CHR$(Y);
27060 IF PR THEN LPRINT CHR$(Y);

```

```

27070 NEXT X: IF F(1) THEN RETURN
27080 IF PR THEN LPRINT
27090 GOTO 27010
27999 'Saven
28000 GOSUB 700
28010 GOSUB 400: BEG=ADRES
28020 GOSUB 400:IF ADRES=0 THEN RETURN
28030 BSAVE "A:"+FI$,BEG,ADRES:REM Bij gebruik van diskdrive
28035 REM BSAVE "cas:"+FI$,BEG,ADRES:REM Bij gebruik van cass
etterecorder
28040 RETURN
28999 'Laden
29000 GOSUB 700
29010 BLOAD FI$
29020 RETURN
29999 'Assembleren disassembly
30000 KEY (1) OFF: A$="."+CO$: GOTO 20023
38000 F(1)=TRUE: RETURN
39999 'Inlezen tabellen
40000 DIM T$(14,8),T(14)
40010 FOR T=0 TO 14: READ T(T)
40020   FOR NR=0 TO 7: READ T$(T,NR)
40030     NEXT NR
40040 NEXT T
40050 DIM ED(7)
40060 FOR T=0 TO 7: READ ED(T): NEXT T
40070 DIM MN$(39),MA(39,1)
40080 FOR T=0 TO 39
40090   READ MN$(T),MA(T,0),MA(T,1)
40100 NEXT T
40200 READ PL: DEFUSR1=PL: GOSUB 40300
40210 READ PL: DEFUSR2=PL: GOSUB 40300
40220 GOTO 41000
40300 XO=0: READ AB
40310 FOR X=PL TO PL+AB: READ A$
40320 BYTE=VAL("&H"+A$)
40330 POKE PL,BYTE: PL=PL+1
40340 XO=XO XOR BYTE: NEXT X
40350 IF XO<>0 THEN PRINT "Data-error in regel 53000 of 53100
": BEEP: END
40360 RETURN
41000 TRUE=-1: FALSE=NOT TRUE
41010 KEY 1,"end"+CHR$(13)
41020 KEY 2,"menu"+CHR$(13)
41030 KEY 3,"->hex"+CHR$(13): HEX=TRUE
41040 KEY 4,"pr.uit"+CHR$(13): PR=FALSE
41050 ON KEY GOSUB 38000: KEY ON
41060 DIM F(10)
41999 RETURN
49999 'Data T,T$
50000 DATA 195,jp V,*, "out (V),a", "in a,(V)", "ex (sp),hl", "ex
de,hl",di,ei
50010 DATA 0,nop, "ex af,af'",djnz V,jr V,ret,exx,jp (hl), "ld
sp,hl"
50020 DATA 2, "(bc),a", "a,(bc)", "(de),a", "a,(de)", "(V),hl", "hl
,(V)", "(V),a", "a,(V)"

```

```

50030 DATA 7,rlca,rrca,rla,rra,daa,cpl,scf,ccf
50040 DATA 128,add,adc,sub,sbc,and,xor,or,cp
50050 DATA 0,rlc,rrc,rl,rr,sla,sra,*,srl
50060 DATA 67,"(V),bc","bc,(V)","(V),de","de,(V)",*,*,"(V),sp",
,"sp,(V)"
50070 DATA 71,"i,a","r,a","a,i","a,r",*,*,*,*
50080 DATA 0,bc,de,hl,af,bc,de,hl,af
50090 DATA 0,nz,z,nc,c,po,pe,p,m
50100 DATA 0,bc,de,hl,sp,*,*,*,*
50110 DATA 0,b,c,d,e,h,l,(hl),a
50120 DATA 0,ld,cp,in,ot,i,d,ir,dr
50130 DATA 0,im 0,im 1,im 2,neg,reti,retn,rld,rrd
50140 DATA 0,0,1,2,3,4,5,6,7
50999 'Data ed
51000 DATA 70,86,94,68,77,69,111,103
51999 'Data mn,mn$
52000 DATA "halt",118,255,H,0,231
52010 DATA "jr P,Z",32,231,"ld Q,Y",1,207
52020 DATA "add hl,Q",9,207,ld I,2,199
52030 DATA inc Q,3,207,dec Q,11,207
52040 DATA inc R,4,199,dec R,5,199
52050 DATA "ld R,X",6,199,J,7,199
52060 DATA "ld R,R",64,192,K R,128,192
52070 DATA ret P,192,199,pop O,193,143
52080 DATA H,201,143,"jp P,Y",194,199
52090 DATA G,195,199,"call P,Y",196,199
52100 DATA push O,197,143,call Y,205,207
52110 DATA K X,198,199,rst W,199,199
52120 DATA "??",0,0
52129 'Cb-codes
52130 DATA L R,0,192,"bit U,R",64,192
52140 DATA "res U,R",128,192,"set U,R",192,192
52150 DATA "??",0,0
52159 'Ed-codes
52160 DATA "in R,(c)",64,199,"out (c),R",65,199
52170 DATA "sbc hl,Q",66,207,"adc hl,Q",74,207
52180 DATA "ld (Y),Q",67,207,"ld Q,(Y)",75,207
52190 DATA ld N,71,231
52200 DATA SS,160,192
52210 DATA T,0,0
52220 DATA "??",0,0
52999 'Code copy-routine
53000 DATA &HBE00,35,2A,30,BE,ED,5B,32,BE,ED,4B,34,BE,78,B1,C
8,ED,52,F5,B7,ED,5A,F1,38,03,ED,B0,C9,09,2B,EB,09,2B,EB,ED,B8
,C9,3B
53099 'Code zoek-routine
53100 DATA &HBE40,53,2A,80,BE,ED,4B,82,BE,78,B1,C8,11,00,BF,1
A,ED,B1,C0,22,80,BE,ED,43,82,BE,3A,86,BE,47,05,28,08,13,1A,BE
,23,20,DB,18,F5,21,FF,FF,22,F8,F7,2A,80,BE,2B,22,84,BE,C9,29

```

Checksums:

Gebruik het programma van paragraaf 4.6.

1:9F59	2:FD2F	3:03DF	10:75CB	20:AB48	
30:BF41	99:E0A9	100:BF4F	110:2559	120:0CE7	
130:B838	140:1854	170:4103	180:699A	190:6CF6	
199:B2A2			blok-checksum :		E870
200:91FC	210:2C15	220:6A76	299:B262	300:12CC	
310:2B9C	320:6B76	399:6C78	400:1710	410:1430	
420:AE71	430:67CB	499:7538	500:19FA	510:6826	
520:7835			blok-checksum :		3BB7
530:1990	540:14D0	550:B871	560:830D	570:57AE	
580:1BA5	590:DFCA	599:0B6C	600:69A0	610:7199	
620:6E76	699:295F	700:1A10	710:148C	720:6F76	
799:8DEC			blok-checksum :		2B1A
800:12F3	810:1CC8	820:9733	830:B3E7	840:16E2	
850:299C	860:7458	899:1869	900:5C38	910:932B	
949:7092	950:71EB	960:AF75	999:6294	1000:1EE6	
1010:9141			blok-checksum :		6120
1020:C723	1030:DC32	1040:2640	1099:990A	1100:22E6	
1110:42AE	1120:6349	1130:CF41	1199:6CAC	1200:28C9	
1210:0182	1220:E9B8	1230:CCF6	1299:CBF1	1300:A391	
1399:CD01			blok-checksum :		52BD
1400:323C	1410:01BE	1420:CE76	1499:0B8E	1500:EF86	
1510:6523	1520:A728	1530:B49C	1540:03E1	1550:D725	
1560:D176	1599:524E	1600:EACB	1610:9997	1620:9E1C	
1630:0C0D			blok-checksum :		1FE0
1640:43C2	1650:7543	1660:7CDC	1670:7544	1680:D376	
1699:933C	1700:97F6	1710:95CE	1720:3A82	1730:4473	
1740:CCC0	1750:D2F6	1799:A7A4	1800:FBC8	1810:E9DA	
1820:D276			blok-checksum :		7313
1999:8EC1	2000:2218	2010:96FF	2099:B1B8	2100:3C6D	
2110:D67F	2120:6F8B	2130:E815	2199:34B6	2200:3B7F	
2210:56BF	2220:C4E0	2299:7264	2300:A2D0	2310:8A49	
2320:8456			blok-checksum :		0307
2399:CF00	2400:7030	2410:4B0C	2420:13E4	2499:64B9	
2500:EDE4	2510:F057	2520:2086	2530:A70F	2599:A733	
2600:DAB7	2610:3343	2620:4313	2630:0D6C	2640:D321	
2699:D381			blok-checksum :		02EC
2700:5C9B	2710:444F	2720:5397	2730:8CDE	2740:E46D	
2750:F470	2760:96CF	2799:A965	2800:9D97	2810:3486	
2820:1A26	2899:58D2	2900:4767	2910:5B0C	2913:E67C	
2916:72EC			blok-checksum :		B319
2918:D6F6	2920:F216	2930:FC8F	2940:7DBE	2950:059B	
2960:3BB5	2970:C52C	2999:9E53	3000:83B3	3010:5A58	
3020:875C	3030:6255	3040:3631	3050:842F	3060:3E2A	
3099:099E			blok-checksum :		35D2
3100:B3CF	3110:8CAF	3120:F92A	3130:F6A2	3140:F7D9	
3150:402A	3199:344A	3200:C5ED	3210:83C5	3299:95A4	
3300:7054	3310:9C85	3320:6390	3399:A91F	3400:910A	
3410:A485			blok-checksum :		AFDD
3420:A71F	3499:D525	3500:B97C	3510:6A02	3520:A73F	
3599:6A60	3600:C17E	3699:6AB0	3700:35E2	3799:C048	
3800:5229	4999:E54D	5000:84D6	5010:D1F6	5099:350A	
5100:161E			blok-checksum :		C7DE

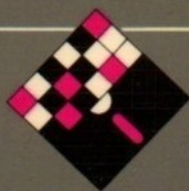
5110:069D	5199:A472	5200:94FE	5210:C092	5299:A692	
5300:3C8E	5310:039F	5320:06ED	5399:B950	5400:9E83	
5410:F0EE	5420:FFE6	5999:635F	6000:35AC	6010:5BA3	
6020:E8E8				blok-checksum :	853C
6030:3FF2	6040:6352	6050:E772	6060:9889	6099:8B8E	
6100:DED7	6110:3224	6120:4056	6130:7DA8	6200:34EE	
6210:15FC	6220:4FDD	6300:AF65	6310:B045	6320:8514	
6330:1A7A				blok-checksum :	F043
6399:6A3A	6400:93E9	6410:8066	6420:E6A3	6430:C10E	
6440:083A	6450:63C3	6460:BA45	6470:D5FA	6480:CA28	
6490:4E35	6499:DA72	6500:5844	6510:94D7	6520:D0F0	
6530:856C				blok-checksum :	E501
6540:C9C7	6600:D976	6999:823A	7000:F127	7005:C9E2	
7010:FF01	7020:0D28	7030:954E	7040:5466	7050:27B2	
7060:D485	7070:04D5	7080:D976	7099:D895	7100:D8B4	
7109:A3BF				blok-checksum :	5463
7110:1281	7120:05A6	7130:CBC5	7140:5DB7	7199:D3D8	
7200:B29B	7210:AE9D	7220:3214	7240:D976	9999:727C	
10000:8F94	10010:3907	10020:A389	10030:E880	10040:190A	
10050:B4FD				blok-checksum :	C865
10060:63DD	10070:5205	10080:6595	10090:5FD0	10100:9CE0	
10110:51FD	10120:4FBD	10999:1368	11000:F142	11005:9969	
11010:C542	11015:C9ED	11020:337E	11030:C485	11040:B03D	
11050:1D9C				blok-checksum :	FC43
11055:58B0	11060:90AD	11065:F0C0	11070:4A8E	11080:F571	
11090:878B	11100:8E77	19999:61CF	20000:0508	20010:5A83	
20020:F84F	20023:2D01	20027:6A53	20030:8107	20040:AAE9	
20050:2FBA				blok-checksum :	24E2
20060:1928	20070:D821	20075:A24C	20080:089D	20090:A859	
20100:CA4B	20110:4702	20120:8B08	20130:1081	20140:A60C	
20150:1319	20160:F003	20170:BE98	20180:44E4	20190:6677	
20195:A95E				blok-checksum :	D51C
20199:52A7	20200:C8D0	20210:BB5E	20220:5119	20299:7835	
21000:A13E	21020:8D0C	21030:3A83	21040:CF9E	21050:5BDA	
21060:E75E	21070:EA28	21080:54F0	21090:ABC4	21100:8B68	
21999:D8B5				blok-checksum :	76A2
22000:668B	22010:5101	22020:5105	22030:A0BF	22999:0209	
23000:CD49	23010:4A31	23020:7E5D	23030:BE71	23040:9777	
23999:21F6	24000:53EA	24010:5C6A	24020:9877	24999:4B36	
25000:26B4				blok-checksum :	B0CF
25010:07D5	25020:9D37	25030:5B57	25040:688F	25050:2814	
25060:3373	25070:9AFD	25080:C1DD	25090:4861	25100:CB8C	
25999:BF00	26000:4B8A	26010:F368	26020:BECE	26030:1D3E	
26040:9A25				blok-checksum :	0002
26050:2580	26060:A297	26070:ABE8	26080:82C2	26090:F78E	
26100:F3CE	26110:2A29	26120:32BC	26130:ABE4	26140:68D7	
26150:EF8E	26999:DE51	27000:4B9A	27010:F36A	27020:BFCE	
27030:001F				blok-checksum :	788D
27040:E38D	27050:2801	27060:06B4	27070:4646	27080:82C4	
27090:0797	27999:A9E9	28000:7A72	28010:9427	28020:1BE7	
28030:1149	28035:708D	28040:A177	28999:AB31	29000:8272	
29010:7FE4				blok-checksum :	08AC
29020:A277	29999:D896	30000:0CAA	38000:2BF0	39999:F9DF	
40000:5840	40010:DB91	40020:19D7	40030:988F	40040:CC64	
40050:5F8D	40060:2528	40070:6F16	40080:A834	40090:A49D	
40100:CBE4				blok-checksum :	B845



40200:8785	40210:E785	40220:CB84	40300:541F	40310:C5EA	
40320:9941	40330:9707	40340:E2BF	40350:53E3	40360:9D77	
41000:9615	41010:F4A5	41020:CBDB	41030:26F4	41040:028B	
41050:6E62					blok-checksum : 8F39
41060:68D3	41999:A937	49999:A7F9	50000:9E2E	50010:847D	
50020:BBDE	50030:50D0	50040:AA30	50050:983D	50060:7AAD	
50070:E510	50080:44B2	50090:EBF9	50100:A20D	50110:4F84	
50120:6DCA					blok-checksum : 2120
50130:2E44	50140:3F8F	50999:4E25	51000:AD61	51999:2809	
52000:AE49	52010:5763	52020:6E02	52030:40E0	52040:A4DA	
52050:610C	52060:2370	52070:E692	52080:84C5	52090:79AA	
52100:6382					blok-checksum : 8114
52110:4532	52120:BACF	52129:F521	52130:0A2F	52140:F76F	
52150:1AD0	52159:02A2	52160:D039	52170:66F1	52180:E152	
52190:3052	52200:5D67	52210:FD94	52220:FACF	52999:9910	
53000:4FE1					blok-checksum : 0713
53099:2CD9	53100:DFB4				blok-checksum : 6775







PROFESSIELE  
SOFTWARE REEKS

## Software voor zakelijke en technische toepassingen

De bezitter van een MSX computer vindt in dit boek een interessante verzameling programma's, die zich richten op een professioneel gebruik.

Een uitgebreid scala van algemeen wiskundige toepassingen, statistische en financiële berekeningen, staat u ten dienste. Daarnaast worden programma's gegeven voor:

- tekstverwerking en databestanden;
- het componeren van muziekstukjes;
- het tekenen op het grafisch scherm;
- het maken van calculaties;
- het ontwikkelen van shapes;
- het analyseren van geluid;
- het simuleren van digitale schakelingen en
- een uitgebreide versie van THTSIM, het bekende simulatieprogramma dat op de TH Twente werd ontwikkeld.

Wie zich thuisvoelt in de wereld van de machinecode zal veel gemak ondervinden van het programma voor assembleren en disassembleren. Tenslotte zijn er nog enkele spelletjes opgenomen, want zelfs in de professionele wereld is er wel een moment voor ontspanning te vinden.

ISBN 90-6789-051-0



**ADDISON-WESLEY**



**UITGEVERIJ OMIKRON**