

MSX

LEERBOEK

OPDRACHTEN BIJ

DEEL 1

WESSEL AKKERMANS/PIET DEN HEIJER

Vragen en opdrachten

1 Standaard Hardware en Software

1-1:

Wat betekent de term "hardware"?

1-2:

Wat is software?

1-3:

Waardoor ontstaan er steeds nieuwe BASIC-dialecten?

1-4:

Noem drie belangrijke, in MSX gestandaardiseerde, gebieden.

1-5:

Op welke manier kunnen we aan de computer kenbaar maken dat we een compleet gegeven hebben ingetoetst?

1-6:

Noem twee gestandaardiseerde chips die in alle MSX-computers worden gebruikt.

1-7:

Waarom worden met MSX-computers maximaal 40 tekens per regel op het beeldscherm weergegeven?

1-8:

Wat is een monitor en welke twee soorten zijn er?

1-9:

Noem drie soorten afdrukeenheden.

1-10:

Welke soort printer kan onmogelijk alle MSX-tekens afdrucken?

1-11:

Hoe wordt de standaard, volgens welke printers op MSX-computers worden aangesloten, genoemd?

1-12:

Hoeveel cijfers kent het binaire stelsel en welke cijfers zijn dat?

1-13:

Hoeveel verschillende combinaties zijn er te maken met vier binaire cijfers en welke combinaties zijn dat?

1-14:

Hoeveel bytes is 16 kbytes?

1-15:

Noem twee soorten interne geheugens.

1-16:

Noem twee soorten externe geheugens.

1-17:

Welk soort geheugen verliest zijn inhoud zodra de spanning wordt uitgeschakeld?

2 Voorbereiding tot het programmeren

2-1:

Wat is een vertaalprogramma?

2-2:

Welke twee soorten vertaalprogramma's zijn er?

2-3:

Wat is het kenmerkende verschil tussen de twee soorten vertaalprogramma's?

2-4:

Wat is een variabele?

2-5:

Geef een ander woord voor "object code".

2-6:
Hoeveel BASIC-statements passen er op 1 programmaregel?

2-7:
Noem de drie elementen waaruit de taal BASIC is opgebouwd.

2-8:
Wat is een commando?

2-9:
Geef het verschil aan tussen directe en indirecte mode.

2-10:
Wat zijn de vier stappen die tot een goed werkend programma leiden?

2-11:
Hoe is de stroomrichting in een stroomdiagram?

2-12:
Wat is een syntax-fout?

3 Programma's invoeren en corrigeren

3-1:
Wat is het verschil tussen een commando en een programma-statement?

3-2:
Wat doet het commando NEW?

3-3:
Met welk commando kan de regelnummering van een programma zo worden gemaakt, dat het laagste regelnummer 200 wordt, en dat iedere volgende regel een waarde 25 hoger is dan de vorige?

3-4:
Geef de eerste 5 regelnummers van het programma dat is hernummerd met het volgende commando: RENUM 500,10,3

3-5:

Tik de volgende regel in:

```
1000 PRINT "testregel"
```

Geef nu het commando LIST en wijzig de regel, zonder de reeds bestaande tekst opnieuw in te tikken, als volgt:

```
1000 PRINT "dit is een testregel"
```

3-6:

Wat gebeurt er nadat het commando AUTO is ingegeven?

3-7:

Wat betekent het commando AUTO 15,3?

3-8:

Wat doet de computer als je, nadat het commando uit vraag 3-7 is uitgevoerd, het commando AUTO 10 geeft?

3-9:

Met welk commando kunnen de in het programmeergeheugen staande programmeerregels op het beeldscherm worden afgedrukt?

3-10:

In welke situatie is het commando LIST. zeer nuttig?

4 Werken met constanten en variabelen

4-1:

Geef van de volgende constanten aan of het numerieke dan wel alfanumerieke constanten zijn:

```
123
```

```
"123"
```

```
-123
```

```
"ABC"
```

4-2:

Geef van de volgende variabelen aan of het numerieke dan wel alfanumerieke variabelen zijn:

```
A$
```

```
NAAM
```

```
SOM$
```

```
B%
```

4-3:

Geef een voorbeeld van de volgende soorten constanten:

- Gehele getallen (integers).
- Constante met vaste komma.
- Constante met drijvende komma.

4-4:

Welke soort constanten worden er met de volgende aanduidingen aangegeven:

%, # en !

4-5:

Hoeveel is &H100 in het decimale talstelsel?

4-6:

Hoeveel is &B100 in het decimale talstelsel?

4-7:

Wat wordt er op het scherm afgedrukt als de volgende BASIC-statements worden uitgevoerd:

```
PRINT 28
PRINT HEX$(10)
PRINT STR$(10)
PRINT VAL(STR$(10))
PRINT BIN$(8)
```

4-8:

Van welk type is variabele F, nadat het statement DEFINT A-Z is uitgevoerd?

4-9:

Met welk statement kan een variabele tot het type "enkelvoudige nauwkeurigheid" worden gedeclareerd?

4-10:

Is het volgende programma correct? Indien het niet correct is, wat moet er dan worden gewijzigd?

```
10 DEFSTR A,D
20 A="tekst 1"
30 B="tekst 2"
40 PRINT A;B
```

4-11:

Indien regel 10 uit het programma van vraag 4-10 helemaal zou worden weggelaten, hoe zouden dan de overige regels moeten luiden, om de twee teksten op het beeldscherm afgedrukt te krijgen?

5 Het toekennen van waarden aan variabelen

5-1:

Wat is de functie van het REM-statement?

5-2:

Is de volgende programmaregel goed? Indien niet, hoe zou die dan moeten luiden?

```
10 REM waardetoeckenning: LET A=10
```

5-3:

Schrijf een programma waarin aan de variabelen X en Y een decimale waarde wordt toegekend en waarin die variabelen bij elkaar worden opgeteld. Zet het resultaat van de optelling in variabele Z.

5-4:

Wat is het type van de variabelen uit het programma van vraag 5-3 en hoe kunnen we er zorg voor dragen dat alle variabelen in dat programma van het type "integer" worden?

5-5:

Wat is (wanneer de computer dat uitrekent) de uitkomst van:

```
8 + 3 - 6 * 2
(8 + 3 - 6) * 2
2 * 2 ^ 3
(2 * 2) ^ 3
```

5-6:

Schrijf een programma, waarmee de computer vraagt om een getal in te tikken. Dit getal moet vervolgens in een variabele worden gezet. Daarna moet het programma het kwadraat van het ingetikte getal op het beeldscherm afdrucken.

5-7:

Wat is er niet goed aan het volgende statement?

```
100 LET A="123"
```

5-8:

Wat wordt er op het beeldscherm afgedrukt met het volgende programma?

```
100 A$="ABR"  
110 B$="ACADA"  
120 C$="BRA"  
130 P$=A$+B$+C$  
140 PRINT P$
```

5-9:

Wat is het resultaat van de volgende commando's?

```
PRINT BIN$(32)  
PRINT VAL(BIN$(32))  
PRINT VAL("&B"+BIN$(32))
```

5-10:

Schrijf een programma, waarmee een getal wordt opgevraagd. Het resultaat van het programma moet er als volgt uitzien:

```
getal? xx  
xx keer xx = yy
```

5-11:

Schrijf een programma waarmee in een INPUT-statement drie variabelen (Lengte, Breedte en Hoogte) worden opgevraagd en waarmee de berekende inhoud op het beeldscherm wordt afgedrukt.

5-12:

Schrijf een programma, waarmee een tekst van 5 tekens wordt opgevraagd en opgeslagen in variabele P\$. De tekens mogen tijdens het intikken niet op het beeldscherm worden afgedrukt. Hoe kun je na uitvoering van het programma controleren welke tekens er in P\$ staan?

5-13:

Schrijf een programma waarmee twee variabelen (X en Y) met waarden, die via het toetsenbord worden ingevoerd, worden geladen. Zorg vervolgens dat (binnen het programma) X de waarde van Y krijgt en Y de waarde van X, doch zonder het SWAP-statement te gebruiken.

5-14:

Schrijf een programma dat hetzelfde doet als het programma uit vraag 5-13, doch gebruik nu het SWAP-statement.

5-15:

Schrijf een programma waarmee een tekst, die via het toetsenbord wordt ingegeven, in een variabele wordt gezet. In de via het toetsenbord in te geven tekst komen komma's voor.

5-16:

Schrijf een programma, waarmee tekens, die via het toetsenbord worden ingetikt, onmiddellijk op het beeldscherm worden afgedrukt. In dit programma komt het volgende statement voor:

```
IF I$="" GOTO <regelnr>
```

6 Afdrukken van gegevens

6-1:

Welke twee beeldschermmoden kent MSX-BASIC?

6-2:

Wat is het aantal regels en het aantal tekens per regel in de SCREEN 1 mode?

6-3:

Wat is in de SCREEN 2 mode het aantal pixels (horizontaal en verticaal)?

6-4:

Schrijf een programma waarmee in de SCREEN 3 mode de volgende tekst wordt afgedrukt:

```
MSX  
BASIC
```

Gebruik hierbij de statements OPEN, PRINT en DRAW. De tekst moet in het midden van het beeldscherm komen te staan.

6-5:

Op welke vier manieren kan het beeldscherm worden schoongemaakt?

6-6:

Met welk statement kan de regellengte worden veranderd?

6-7:

Wat is de regellengte (default-waarde) bij het inschakelen van de computer voor de SCREEN 0 en 1 mode?

6-8:

Wat is de maximale regellengte voor respectievelijk de SCREEN 0 en 1 mode?

6-9:

Schrijf een programma waarmee achtereenvolgens:

- a) de beeldschermmode 0 wordt ingesteld.
- b) de regellengte op 40 wordt ingesteld.
- c) met behulp van INPUT-statements wordt gevraagd naar naam, adres en woonplaats.
- d) twee lege regels worden afgedrukt.
- e) met behulp van PRINT-statements de ingevoerde naam, adres en woonplaats worden afgedrukt.

6-10:

Wat is de taak van de functie TAB en in combinatie met welk statement wordt de functie TAB altijd gebruikt?

6-11:

Wat is de waarde van de meest linker positie van een regel en wat mag de waarde van het argument X in de functie TAB(X) zijn?

6-12:

Verander het programma uit vraag 6-9 zodanig, dat met het afdrukken van naam, adres en woonplaats steeds op positie 6 van de respectievelijke regels wordt begonnen. Gebruik hierbij de functie TAB.

6-13:

Wat is de betekenis van X,Y en Z in het statement LOCATE X,Y,Z?

6-14:

Hoe kan, in de indirecte, mode de cursor onzichtbaar worden gemaakt?

6-15:

Verander het programma uit paragraaf 6.7 zodanig, dat de omkaderde tekst "MSX-BASIC" in het midden van het beeldscherm komt te staan. Gebruik hiervoor LOCATE-statements (regelnummers 125,135,145,155 en 165). Verwijder door middel van het programma ook de functienamen op de laatste regel van het beeldscherm.

6-16:

Wat is de betekenis van de functie SPC en met welk statement wordt de functie altijd gebruikt?

6-17:

Schrijf een programma waarmee achtereenvolgens:

- a) de regellengte op 40 wordt ingesteld.
- b) op de bovenste regel in volgorde de letters A tot en met D worden afgedrukt. De letter A moet hierbij op positie 7 worden afgedrukt en de tussenruimte tussen elk paar letters moet 7 posities zijn. Gebruik hiervoor de functie SPC.
- c) voer punt b) ook uit voor de op een na laatste regel van het beeldscherm.

6-18:

Los het probleem, gesteld in vraag 6-17, ook op met de functie SPACE\$(X).

6-19:

Welke twee codetabellen kunnen voor MSX-BASIC worden onderscheiden?

6-20:

Wat is de code (decimale waarde) voor de letter P?

6-21:

Druk in de indirecte mode de letter P af. Gebruik hierbij de functie CHR\$ in combinatie met het PRINT-statement.

6-22:

Maak een tabel met de 26 letters A tot en met Z en de 10 cijfers 0 tot en met 9, met hun bijbehorende codes in decimale- en hexadecimale waarden.

6-23:

Druk, in de indirecte mode, door middel van het optellen van codes (CHR\$(X)) uw naam af op het beeldscherm.

6-24:

Wat is de functie van het PRINT USING statement?

6-25:

Hoe ziet het PRINT-masker eruit, wanneer in een kolom gehele getallen met een maximale waarde van 2500 moeten worden afgedrukt.

6-26:

Schrijf een programma waarbij alle getallen, van 50 tot en met 1250 en oplopend met 100 (50, 150, 250, enz.), worden afgedrukt. Gebruik hierbij het PRINT USING statement.

6-27:

Hoe ziet het PRINT-masker eruit, wanneer het gewenste aantal cijfers voor de komma 4 is en achter de komma 2.

6-28:

Hoe worden de volgende getallen afgedrukt bij gebruikmaking van het PRINT-masker "###.##".

```
127.1
 24
32.628
114.914
2014.6      (Wees op uw hoede!)
```

6-29:

Hoe kan men in een PRINT-masker aangeven, dat bij positieve getallen een plusteken (voor het getal) en bij negatieve getallen een minteken moet worden afgedrukt.

6-30:

Schrijf een programma waarmee alle getallen van -110 tot en met 115, oplopend met 15, in een kolom worden afgedrukt. De positieve getallen moeten hierbij voorzien zijn van een plusteken en de negatieve getallen van een minteken (voor de getallen).

7 Sprongen

7-1:

Welke sprongopdrachten kunnen worden onderscheiden en wat is het onderlinge verschil?

7-2:

Welke foutboodschap zal door de computer worden afgedrukt, wanneer in het programma in paragraaf 7.1 op regelnummer 90 abusievelijk GOTO 15 is geplaatst?

7-3:

Wat is het formaat van een IF...GOTO statement?

7-4:

Welke vergelijkingen zijn mogelijk in een IF...GOTO statement?

7-5:

Hoe gaat het IF...GOTO statement eruit zien, wanneer bij $F \leq 45$ naar regelnummer 120 moet worden gesprongen?

7-6:

Schrijf een programma, waarmee een tabel wordt afgedrukt, waarin in de ene kolom de temperaturen in graden Fahrenheit staan vermeld en in de andere kolom de bijbehorende temperaturen in graden Celcius. De tabel moet starten bij 32 graden Fahrenheit en eindigen bij 45 graden Fahrenheit. Verder moet de tabel oplopen met 1 graad Fahrenheit. Gebruik voor het afdrukken het PRINT USING statement.

7-7:

Verander het programma uit vraag 7-6 zodanig, dat de begin- en eindwaarde door middel van INPUT-statements via het toetsenbord kunnen worden ingevoerd.

7-8:

Wat is een menu?

7-9:

Welk statement wordt gebruikt om vanuit een menu naar de gekozen routine te springen?

7-10:

Schrijf een programmadeel, waarbij het volgende menu wordt afgedrukt:

OPPERVLAKTE BEREKENINGEN

1. Oppervlakte rechthoek
2. Oppervlakte cirkel
3. Oppervlakte driehoek
4. Programma beëindiging

Opmerkingen:

- a) De regellengte moet worden ingesteld op 40.
- b) De inhoud van de functietoetsen (regel 24) moeten worden verwijderd.
- c) Tussen "OPPERVLAKTE BEREKENINGEN" en het menu moeten 2 blanco regels worden afgedrukt en tussen de punten uit het menu onderling 1 blanco regel.
- d) Plaats het menu met het statement LOCATE in het midden van het scherm.
- e) Begin het programma met regelnummer 100 en laat de regelnummers oplopen met 10.

7-11:

Vervolg het programmadeel uit vraag 7-10 met:

- a) Het kiezen vanuit het menu (INPUT-statement). Plaats de vraag "Uw keuze" op regel 23 van het beeldscherm.
- b) Het controleren of de keuze niet kleiner is dan 1 en niet groter dan 4.
- c) Het beëindigen van het programma met het statement END, indien de keuze "4" is.
- d) Het aanroepen, met het statement ON...GOTO, van de gekozen routine. De routines starten respectievelijk op de regelnummers 300, 450 en 600.

7-12:

Schrijf de routines voor de punten 1 tot en met 3 uit het menu van vraag 7-10. Voer hiervoor met INPUT-statements de volgende gegevens in:

punt 1: rechthoekszijden A en B ($O=A*B$)

punt 2: straal R ($O=\pi*R^2$)

punt 3: basis B en hoogte H ($O=0,5*B*H$)

Voer na het invoeren van de gegevens de berekening uit en druk vervolgens het resultaat af. Spring na het uitvoeren van een routine terug naar de statements $A\$=INKEY\$:IF A\$="" GOTO X$, welke volgt op het ON...GOTO statement.

7-13:

Waarvoor dienen subroutines?

7-14:

Met welke statements kan een subroutine worden aangeroepen?

7-15:

Welk statement wordt gebruikt, wanneer vanuit een menu naar een subroutine moet worden gesprongen?

7-16:

Met welk statement kan worden teruggekeerd (terugsprongen) naar het programma, dat de subroutine heeft aangeroepen?

7-17:

Maak van de routines uit vraag 7-12 subroutines en pas ook het programmadeel uit vraag 7-11 aan (ON...GOSUB).

8 Lussen en opslaan

8-1:

Welke logische operators kent MSX-BASIC?

8-2:

Maak de waarheidstabel voor de logische operator OR.

8-3:

Hoe schrijft men het IF...THEN statement, wanneer men $A=A+4$ alleen wil laten uitvoeren, indien $X=2$ en $Y<6$.

8-4:

Hoe schrijft men het IF...THEN statement, wanneer in het geval van $X=3$ of $Y>4$ de berekening $A=A+2$ moet worden uitgevoerd en in het geval de logische uitdrukking niet waar is, de berekening $A=A+4$.

8-5:

Schrijf een programma welke het product van 5 opeenvolgende getallen berekent (bijv. $6*7*8*9*10$). Het eerste getal van de rij moet worden ingevoerd met een INPUT-statement. Gebruik voor het uitvoeren van de berekening

een FOR...NEXT lus. Het ingevoerde getal is dan de beginwaarde en het ingevoerde getal plus 4 de eindwaarde in het FOR...NEXT-statement.

8-6:

Schrijf hetzelfde programma als gesteld in vraag 8-5, maar voer nu ook de eindwaarde in met een INPUT-statement.

8-7:

Schrijf een programma waarmee alle even getallen van X tot en met Y worden opgeteld. Voer X en Y in met INPUT-statements. Voorwaarde is, dat X en Y even getallen zijn. De stapwaarde is 2.

8-8:

Wat is het resultaat van:

```
INT(2.33)
INT(-12.44)
INT(37)
```

8-9:

Schrijf een programma dat tussen de grenzen X en Y priemgetallen opzoekt en afdruckt. De stappen zijn:

1. Voer X en Y in (INPUT)
2. Maak D=2
3. Indien X deelbaar is door D, ga dan naar 7
Zo niet, ga dan naar 4.
4. Verhoog D met 1.
5. Is D kleiner of gelijk aan SQR(X), ga dan naar 3. Is dit niet zo, ga dan naar 6.
6. Druk X af (priemgetal).
7. Verhoog X met 1.
8. Beeindig het programma indien X groter is dan Y; ga anders naar 2.

Opmerking:

De vierkantswortel uit een getal is nog niet behandeld, doch de functie hiervoor is SQR(X).

8-10:

Wat is het verschil tussen CSAVE en SAVE?

8-11:

Schrijf een programma voor het berekenen van de formule $x=(a*b-c)/2.5$. De variabelen a, b en c moeten worden ingevoerd met INPUT-statements. Het resultaat x moet worden afgedrukt. Het programma behoeft de berekening maar eenmaal uit te voeren. Eindig het programma met END. Start het programma met regelnummer 100 en laat de regelnummers met 10 oplopen.

8-12:

Schrijf het programma uit vraag 8-11 onder de naam "BER1" op cassette met het commando SAVE. Voer hierna het commando NEW uit.

8-13:

Schrijf een programma voor het berekenen van de formule $y=\text{SQR}(d+e)$. De variabelen d en e moeten worden ingevoerd met INPUT-statements. Het resultaat y moet worden afgedrukt. De berekening behoeft maar eenmaal te worden uitgevoerd. Start het programma met het regelnummer van het END-statement uit vraag 8-11. Laat de regelnummering met 10 oplopen.

8-14:

Schrijf het programma uit vraag 8-13 onder de naam "BER2" op cassette met het commando SAVE. Voer hierna het commando NEW uit.

8-15:

Schrijf een programma, waarmee met de variabelen x en y uit de vragen 8-11 en 8-13 de volgende berekening wordt gemaakt: $z=(x-y)/2$.

Het resultaat z moet worden afgedrukt. Start het programma met het regelnummer dat volgt op het regelnummer van het END-statement uit vraag 8-13. Laat de regelnummering met 10 oplopen.

8-16:

Combineer het programma uit vraag 8-15 met het programma "BER1", dat op cassette staat. Gebruik hiervoor het commando MERGE.

8-17:

Combineer ook het programma "BER2", dat op cassette staat met het programma in het geheugen. Gebruik ook hiervoor weer het commando MERGE.

8-18:

Verwijder regelnummers 180 en 250 en test de "gemergde" programma's uit de vragen 8-16 en 8-17 uit.

8-19:

Schrijf een programma waarmee een cassetteband wordt teruggespoeld (Deze opdracht kan alleen worden uitgevoerd wanneer er een cassetterecorder met remote control ter beschikking is.). De stappen zijn:

1. Na het instellen van de cassetterecorder op "terugspoelen", moet als gevolg van het indrukken van de RETURN-toets (CHR\$(13)) de motor worden ingeschakeld.
2. Wanneer de band volledig is teruggespoeld, moet als gevolg van het indrukken van de RETURN-toets de motor weer worden uitgeschakeld.
3. Transporteer de band naar het begin. De tijd, die hiervoor nodig is, kan worden bereikt met een FOR...NEXT lus. Test zelf hoeveel malen de lus moet worden doorlopen. Gedurende de tijd, dat de FOR...NEXT lus wordt doorlopen, moet de motor zijn ingeschakeld. Het inschakelen van de motor geschiedt door het indrukken van de RETURN-toets. Het uitschakelen van de motor geschiedt automatisch na het verlaten van de FOR...NEXT lus.

9 Tijd en fouten

9-1:

Schrijf een programma, waarmee een tekst op het beeldscherm wordt afgedrukt. Na 4 seconden moet de tekst verdwijnen, doordat er een CLS-statement wordt uitgevoerd.

9-2:

Wat is het verschil tussen het INTERVAL OFF en het INTERVAL STOP statement?

9-3:

Schrijf een programma, waarmee een tekst via het toetsenbord moet worden ingegeven. Indien de ingegeven tekst 40 tekens of korter is, dan moet deze op het beeldscherm worden afgedrukt. Is de tekst langer dan 40

tekens, dan moet de boodschap "Past niet op de regel" worden afgedrukt.

9-4:

Schrijf een programma, waarmee een tekst, die via het toetsenbord wordt ingevoerd en die korter is dan 40 tekens, gecentreerd op het beeldscherm wordt afgedrukt. Dit wil zeggen, dat indien de tekst bijvoorbeeld 10 tekens lang is, de eerste letter op horizontale positie 16 moet beginnen. Zou de tekst 30 tekens lang zijn, dan zou de eerste letter op horizontale positie 6 moeten beginnen.

9-5:

Schrijf een programma, waarmee een aantal letters aan een variabele worden toegekend. Vervolgens moet de eerste letter uit die string naar de laatste plaats in die string worden verplaatst, terwijl alle overige letters een plaatsje naar links verschuiven. Voorbeeld: "abcde" moet uiteindelijk "bcdea" worden.

9-6:

Schrijf een programma dat hetzelfde doet als het programma uit vraag 9-5, doch nu met een string die via het toetsenbord wordt ingevoerd.

9-7:

Schrijf een programma, waarmee een via het toetsenbord ingevoerde tekst op het beeldscherm wordt afgedrukt. Na het afdrukken dient het programma de eerste letter van de tekst op de laatste plaats daarvan te zetten en de tekst opnieuw af te drukken. Dit dient zo door te gaan tot het programma wordt gestopt met de CTRL+STOP-toetsen. Bouw in de te doorlopen lus een vertraging in, bijvoorbeeld FOR I=1 TO 100: NEXT I. Het resultaat zal dan een lichtkrant zijn.

9-8:

Pas het programma uit vraag 9-7 zo aan, dat de tekst netjes in het midden van het scherm verschijnt. De tekst mag in lengte variëren van 2 tot 40 tekens.

9-9:

Schrijf een programma, dat eerst vraagt om een tekst in te tikken en daarna vraagt van welke letter moet worden geteld hoe vaak die voor komt in de ingegeven tekst.

9-10:

Voer het volgende programma in:

```
20 A=1: B=2
30 A+B
40 PRINT A;B
50 END
```

Welke foutboodschap verschijnt er na het starten van dit programma en op welke regel heeft die boodschap betrekking?

9-11:

Pas het programma uit vraag 9-10 nu zodanig aan, dat na het starten de volgende informatie over de fout wordt gegeven:

```
Het regelnummer van de regel met de fout.
De foutcode van de fout.
De complete regel waarin de fout optrad.
```

9-12:

Corrigeer, na uitvoering van het in vraag 9-11 geschreven programma, regel 30 als volgt:

```
30 A=A+B
```

Maak hierbij gebruik van de full screen editor, zonder eerst een listing van het programma te maken. Ook is het niet toegestaan de regel helemaal opnieuw in te tikken.

10 Het programmeren van functietoetsen

10-1:

Wat gebeurt er wanneer functietoets F5 wordt ingedrukt?

10-2:

Met welk statement kunnen we een volledige lijst met inhouden van de 10 functietoetsen op het scherm afdrucken?

10-3:

Met welk statement kunnen we de afgedrukte definities van de functietoetsen, op regel 24, weghalen?

10-4:

Tik het volgende programma in:

```
10 PRINT TAB(11);"FUNCTIETOETSEN"  
20 PRINT TAB(10);"ZIJN ZEER NUTTIG"  
30 END
```

Voer na het intikken van het programma de volgende opdrachten uit:

- a) Druk functietoets F5 in.
- b) Voer in de indirecte mode KEY OFF uit.
- c) Voer in de indirecte mode KEY LIST uit.
- d) Druk functietoets F10 in.

Wat is het verschil tussen wat er gebeurt bij het indrukken van functietoets F5 en functietoets F10?

- e) Voer in de indirecte mode KEY ON uit.

10-5:

Met welk statement kan de inhoud van een functietoets worden veranderd?

10-6:

Geef functietoets F2 een dusdanige inhoud, dat eerst KEY OFF wordt uitgevoerd en vervolgens het beeldscherm wordt schoongemaakt.

10-7:

Geef functietoets F3 een dusdanige inhoud, dat eerst KEY ON wordt uitgevoerd en vervolgens het beeldscherm wordt schoongemaakt.

10-8:

Met behulp van welk statement wordt de mogelijkheid geschapen om bij het indrukken van een functietoets naar een subroutine te springen?

10-9:

Met welk statement wordt een functietoets geactiveerd?

10-10:

Schrijf een programma, waarmee na het indrukken van functietoets F4 naar een subroutine wordt gesprongen, die de inhoud van een cirkel berekent. Straal r moet

hierbij met een INPUT-statement worden ingevoerd. Start het programma met regelnummer 100 en laat de regelnummering met 10 oplopen. Start de subroutine met regelnummer 250. Definieer de inhoud van functietoets F4 als "CIRKEL". (De inhoud van een cirkel wordt berekend met de formule: $3.14 * R^2$.)

10-11:

Hoe kan men de tekst van een functietoets van het scherm verwijderen? Verwijder de tekst van functietoets F3.

10-12:

Wat is de functie van het statement:

KEY(functietoets)STOP?

10-13:

Hoe kan men een programma tijdelijk stoppen?

10-14:

Hoe kan men een programma beëindigen?

11 Werken met vaste gegevens

11-1:

Wat is de functie van het statement READ?

11-2:

Is het toegestaan dat er in een en dezelfde DATA-regel zowel numerieke- als alfanumerieke constanten staan?

11-3:

Wat gebeurt er wanneer na het lezen van het laatste element uit een DATA-regel weer een READ-statement wordt uitgevoerd? (Opmerking: Er zijn twee mogelijkheden.)

11-4:

Hoe kunnen we de pointer weer terugzetten naar het eerste element van de eerste DATA-regel?

11-5:

Schrijf een programma waarmee het rekenkundige gemiddelde van een aantal getallen in een DATA-regel wordt berekend. Plaats in de DATA-regel 8 willekeurige getallen.

Lees de getallen uit de DATA-regel in een FOR...NEXT-lus en tel deze ook in de lus bij elkaar op. Na het verlaten van de FOR...NEXT-lus moet het gemiddelde worden berekend.

11-6:

Schrijf een programma waarmee het gemiddelde cijfer van een bepaald schoolvak wordt berekend. Start het programma met het schoonmaken van het beeldscherm en het nul maken van de numerieke variabele, die wordt gebruikt voor de som van alle cijfers. Plaats hierbij in drie DATA-regels 10 willekeurige namen van studenten. De DATA-regels moeten aan het einde van het programma worden geplaatst.

Lees in een FOR...NEXT-lus steeds 1 naam uit de DATA-regels en druk deze af. Plaats achter de naam, met een INPUT-statement het bijbehorende cijfer. Tel in de lus ook de cijfers bij elkaar op. Bereken na het verlaten van de lus het gemiddelde cijfer en druk dat ook af.

Aan het einde van het programma moet een vraagstelling komen of het programma nogmaals moet worden uitgevoerd, zodat het gemiddelde cijfer van een volgend vak kan worden berekend. Denk hierbij aan het terugzetten van de pointer.

12 Opslaan van gegevens

12-1:

Uit hoeveel velden (items) bestaat de array A\$(3)?

12-2:

Schrijf een programma, waarmee vier alfanumerieke gegevens uit een DATA-regel in een 1-dimensionale array (met de naam A) worden gezet. Maak de array niet groter dan absoluut nodig is.

12-3:

Schrijf hetzelfde programma als uit vraag 12-2, doch nu moeten de gegevens niet uit een DATA-regel worden gelezen, maar via het toetsenbord worden ingegeven.

12-4:

Voeg aan het programma uit vraag 12-3 een routine toe, waarmee de gegevens uit de array worden gelezen en op het beeldscherm worden afgedrukt.

12-5:

Waarom dient de functie FRE(0)?

12-6:

Met het statement CLEAR kan de grootte van het string-variabelengeheugen worden bepaald. Geef het statement waarmee de grootte van het string-variabelengeheugen 500 bytes wordt.

12-7:

Met welke functie kan de grootte van het string-variabelengeheugen worden opgevraagd?

12-8:

Waar worden de gegevens uit array A\$ in het geheugen opgeslagen?

12-9:

Schrijf een programma, waarmee een 1-dimensionale array, die uit 11 velden bestaat, wordt gedimensioneerd en waarmee die velden stuk voor stuk worden gevuld met gegevens die via het toetsenbord worden ingegeven. Steeds voordat een gegeven via het toetsenbord wordt ingevoerd, moet op het beeldscherm worden afgedrukt, hoe groot de vrije geheugenruimte is en hoe groot de vrije string-variabelenruimte is.

12-10:

Schrijf een programma, waarmee een 2-dimensionale string-array van 10 rijen en 3 kolommen wordt gemaakt en gevuld via het toetsenbord. Na het vullen van de array moet het programma vragen naar een rij- en kolomnummer, en de inhoud van het opgegeven veld op het beeldscherm afdrucken. Na het afdrucken moet het programma naar een volgend veld vragen.

12-11:

Schrijf hetzelfde programma als in vraag 12-10, doch nu moet de afdrukroutine worden vervangen door een routine waarmee wordt gevraagd in welke kolom moet worden gezocht naar het voorkomen van een op te geven string. Na het opgeven van de kolom en de string, moet het programma met behulp van de functie INSTR zoeken naar het voorkomen van de opgegeven string in de opgegeven kolom, en indien gevonden, dan moeten de rij, de kolom en de inhoud van het veld op het beeldscherm worden afgedrukt. Komt de string niet in de opgegeven kolom voor, dan moet de tekst "<zoekstring> komt niet voor" worden afgedrukt.

12-12:

Indien in een programma een array twee of meer keren wordt gedimensioneerd, welke foutboodschap wordt dan gegeven?

12-13:

Welk statement maakt het mogelijk een array opnieuw te dimensioneren?

12-14:

Schrijf een subroutine waarmee array A\$ opnieuw wordt gedimensioneerd tot een 1-dimensionale array met 21 velden en waarmee de array wordt gevuld met waarden die via het toetsenbord worden ingegeven. Deze subroutine dient te starten op regelnummer 3000. Schrijf deze subroutine als een ASCII-file naar cassette, zodat die later met de functie MERGE kan worden samengevoegd met een programma.

12-15:

Schrijf een subroutine waarmee de inhoud van de array uit vraag 12-14 op het beeldscherm kan worden afgedrukt. De beeldscherm-output moet zodanig zijn, dat er steeds een groepje van 7 velden wordt afgedrukt, waarna het programma wacht op het indrukken van de RETURN-toets, voordat het volgende groepje van 7 velden wordt afgedrukt. De subroutine dient te starten met regelnummer 4000 en als ASCII-file naar cassette te worden geschreven.

12-16:

Schrijf een subroutine waarmee de cassetterecordermotor met behulp van het indrukken van de RETURN-toets aan en uit kan worden geschakeld. De routine moet gebruikersaanwijzingen op het beeldscherm geven, en wel:

RETURN = motor aan/uit
ESC = band staat goed

De bedoeling van deze subroutine is, dat de gebruiker in staat wordt gesteld de cassetterecorder (met remote control) terug te spoelen naar de juiste plaats, om daarna verder te gaan met het programma. Het indrukken van de ESC-toets dient tot gevolg te hebben dat de motor wordt uitgeschakeld en dat wordt teruggesprongen naar het programma dat de subroutine heeft aangeroepen. Deze subroutine dient te beginnen op regelnummer 5000 en dient als ASCII-file naar cassette te worden geschreven.

12-17:

Schrijf een subroutine, waarmee de array, die in vraag 12-14 is gemaakt en gevuld, naar een sequentiele file op cassette wordt geschreven. De gebruiker moet voldoende aanwijzingen krijgen voor het klaar zetten van zijn cassetterecorder en de subroutine die op regel 5000 start (uit vraag 12-16) moet worden aangeroepen. De naam, waaronder de file op cassette wordt gezet, moet via het toetsenbord worden opgevraagd en in het OPEN-statement worden verwerkt. Deze subroutine moet beginnen met regelnummer 1000 en moet als ASCII-file naar cassette worden geschreven.

12-18:

Schrijf een subroutine, waarmee een sequentiele file van cassette wordt gelezen. Deze subroutine dient te beginnen op regelnummer 2000. De ingelezen gegevens moeten in de string-array, die met de subroutine van regelnummer 3000 (vraag 12-14) is gemaakt, worden opgeslagen. Verder gelden voor deze subroutine dezelfde regels als voor de routine uit vraag 12-17. Dit wil onder meer zeggen, dat de subroutine, die op regelnummer 5000 begint, moet worden aangeroepen en dat de naam van de in te lezen file moet worden opgevraagd. Ook deze subroutine dient als ASCII-file naar cassette te worden geschreven.

12-19:

Schrijf een programmadeel, waarmee het volgende menu op het beeldscherm wordt afgedrukt:

- 1 = NIEUWE ARRAY MAKEN
- 2 = ARRAY NAAR CASSETTE SCHRIJVEN
- 3 = ARRAY VAN CASSETTE LEZEN
- 4 = ARRAY OP BEELDSCHERM LIJSTEN
- 5 = PROGRAMMABEEINDIGING

Het programma dient vervolgens de gebruiker te vragen naar zijn keuze en naar de betreffende subroutine te springen. Bij terugkeer van een subroutine dient het programma het menu opnieuw op het beeldscherm af te drukken. Dit programmadeel dient te beginnen op regelnummer 100.

12-20:

Met het programmadeel uit vraag 12-19 in het geheugen dienen nu alle subroutines uit de vragen 12-14 tot en met 12-18 met dit programmadeel te worden samengevoegd. Voeg aan het nu verkregen programma een initialisatieroutine toe (vanaf regelnummer 10), waarin zaken als screen-mode etc. worden geregeld. Voeg bovendien de routine voor het beëindigen van het programma toe. Laat deze beëindigingsroutine starten op regel 6000. SAVE het nu verkregen programma op cassette en test het op goede werking.

----#----

Antwoorden en uitwerkingen

1 Standaard Hardware en Software

1-1:

Alle tastbare delen van een computersysteem.

1-2:

De programma's waarmee de hardware wordt aangestuurd.

1-3:

Door het toepassen van nieuwe hardware componenten moeten nieuwe, speciaal voor die nieuwe componenten ontworpen, software opdrachten worden ontworpen.

1-4:

- 1 - Een aantal componenten binnenin de computer.
- 2 - De verbinding tussen de computer en randapparaten.
- 3 - De programmeertaal: MSX-BASIC.

1-5:

Door het indrukken van de RETURN-toets.

1-6:

De videoprocessor en de geluidsprocessor.

1-7:

Indien er meer dan 40 karakters op een regel zouden worden weergegeven, dan zouden die tekens op een kleurentelevisie niet meer leesbaar zijn.

1-8:

Een monitor is een beeldscherm. Er zijn, gerekend naar de manier van aansluiten, twee soorten, namelijk composite video- en RGB-monitoren.

1-9:

Matrix-, ink jet- en daisy wheel printers.

1-10:

Het letterwiel van de daisy wheel printer heeft slechts 96 tekens, terwijl de MSX-computer 256 verschillende tekens kent.

1-11:

Centronics Parallel Interface.

1-12:

Het binaire stelsel kent slechts 2 cijfers, namelijk de cijfers 0 en 1.

1-13:

Er zijn 16 verschillende combinaties mogelijk, n.l.

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

1-14:

16 maal 1024 bytes = 16384 bytes.

1-15:

ROM = Read Only Memory (leesgeheugen)

RAM = Random Access Memory (lees/schrijfgeheugen)

1-16:

Audiocassetterecorders

Flexibele schijveneenheden.

1-17:

RAM.

2 Voorbereiding tot het programmeren

2-1:

Het programma dat BASIC-statements omzet naar machinetaalinstructies.

2-2:

Compiler en Interpreter.

2-3:

De compiler vertaalt het hele BASIC-programma in 1 keer. Daarna wordt het vertaalde machinetaalprogramma uitgevoerd. Een interpreter vertaalt steeds 1 regel van het BASIC-programma en voert die regel onmiddellijk uit. Daarna wordt de vertaalde regel vergeten en wordt de volgende regel vertaald en uitgevoerd.

2-4:

Een variabele is een vakje in het geheugen, waaraan we een bepaalde naam geven, en waarin we een waarde kunnen plaatsen.

2-5:

Machinetaalprogramma.

2-6:

Dit is afhankelijk van de lengte van iedere BASIC-statement. Het maximum aantal tekens dat op een BASIC-regel past is 255.

2-7:

Statements, functies en commando's.

2-8:

Een commando is een opdracht, waarmee de computer wordt verteld wat deze moet doen met het programma.

2-9:

Indien commando's en statements met regelnummers worden ingegeven, dan zullen ze in de indirecte mode worden uitgevoerd. Dit wil zeggen, pas nadat het programma, waarin ze staan, is gestart. Worden commando's en statements zonder regelnummer ingegeven, dan worden ze onmiddellijk uitgevoerd, ofwel in de directe mode.

2-10:

Probleemstelling, stroomdiagram, programmeren en testen van het programma.

2-11:

Van boven naar beneden en van links naar rechts, tenzij de richting anders is aangegeven met behulp van een pijlpunt in het betreffende lijnstuk.

2-12:

Een fout geschreven statement. Zelfs het kleinste tikfoutje maakt al dat een statement niet meer wordt begrepen door de computer, en dat er een syntax-foutboodschap wordt gegeven.

3 Programma's invoeren en corrigeren

3-1:

Een commando wordt zonder voorafgaand regelnummer ingegeven en direct door de computer uitgevoerd. Een programmaregel wordt altijd voorafgegaan door een regelnummer. Programmaregels, die in het geheugen staan, worden pas door de computer uitgevoerd nadat het commando RUN is gegeven. Bijna alle programma-statements kunnen als commando worden uitgevoerd en bijna alle commando's kunnen als programma-statement worden uitgevoerd.

3-2:

Het commando NEW maakt het programmageheugen helemaal leeg. Ook de in een programma gebruikte variabelen worden uit het geheugen verwijderd.

3-3:

Met het commando RENUM 200,10,25 worden de programmaregels van een programma dat op regelnummer 10 begon zodanig gewijzigd, dat de nieuwe regelnummers beginnen bij 200 en dat iedere volgende regel 25 hoger is dan de vorige.

3-4:

Na RENUM 500,10,3 zullen de eerste 5 programmaregels de volgende nummers hebben: 500, 503, 506, 509 en 512

3-5:

Verplaats de cursor met behulp van de cursor control toetsen naar de eerste letter "t" van het woord "test-regel". Druk daarna op de INS-toets. Tik nu de woorden "dit is een" in en druk dan op de RETURN-toets. Om te controleren of de regel inderdaad naar wens is gewijzigd, kan het commando LIST worden gegeven.

3-6:

Na het commando AUTO, zonder verdere parameters, genereert de computer zelf regelnummers. Wij behoeven dan alleen nog maar de programma-statements in te geven. Steeds nadat we op de RETURN-toets hebben gedrukt, laat de computer het volgende regelnummer zien. De computer zal beginnen bij regelnummer 10 en een verhogingsfactor van 10 gebruiken.

3-7:

AUTO 15,3 houdt in, dat de computer regelnummers genereert, die beginnen bij regelnummer 15. De volgende regel zal steeds een factor 3 hoger zijn dan de vorige. De eerste vijf regelnummers zullen als volgt zijn: 15, 18, 21, 24 en 27.

3-8:

Door het commando AUTO 10, (10 met een komma erachter) te geven, zal de computer beginnen te nummeren bij regelnummer 10. Het volgende regelnummer zal worden verhoogd met dezelfde factor, waarmee de regelnummers bij het vorige AUTO-commando ook werden verhoogd. Was er nog niet eerder een AUTO-commando gegevens, dan zullen de regels steeds met een factor 10 worden verhoogd.

3-9:

Met het commando LIST worden alle in het programmeergeheugen staande programmaregels op het beeldscherm afgedrukt.

3-10:

Het commando LIST. (LIST met een punt er achter) is vooral handig wanneer er een fout in het programma zit die een foutboodschap veroorzaakt. Het commando LIST. drukt dan de regel waarin de fout is opgetreden af op het beeldscherm. Met de "full screen editor" kan de fout dan worden verbeterd. Vervolgens kan het programma opnieuw worden gestart.

4 Werken met constanten en variabelen

4-1:
123 is een numerieke constante.
"123" is een alfanumerieke constante.
-123 is een numerieke constante.
"ABC" is een alfanumerieke constante.

4-2:
A\$ is een alfanumerieke variabele.
NAAM is een numerieke variabele.
SOM\$ is een alfanumerieke variabele.
B% is een numerieke variabele (integer).

4-3:
Gehele getallen, positief of negatief, gevolgd door het %-teken. Voorbeelden: 1234%, -2%.

Reële getallen, positief of negatief, gevolgd door het !- of #-teken. Het !-teken geeft aan dat het een getal met enkelvoudige nauwkeurigheid betreft. Het #-teken geeft aan dat het een getal met dubbelvoudige nauwkeurigheid betreft. Voorbeeld: 45.67#.

Constanten met drijvende komma worden in exponentiele vorm geschreven. Er wordt onderscheid gemaakt tussen enkel- en dubbelvoudige nauwkeurigheid. Voor een constante met drijvende komma en enkelvoudige nauwkeurigheid wordt de letter E als symbool voor de exponent gebruikt. Bij dubbelvoudige nauwkeurigheid wordt de letter D als symbool voor de exponent gebruikt. Voorbeelden: 6.25D9, 1.25E3.

4-4:
% integers.
dubbelvoudige nauwkeurigheid.
! enkelvoudige nauwkeurigheid.

4-5:
De decimale waarde van het hexadecimale getal &H100 is 256. Dit is op de volgende manier te controleren met de computer:

```
PRINT &H100
      256
Ok
```

4-6:

De decimale waarde van het binaire getal &B100 is 4. Dit is op de volgende manier te controleren met de computer:

```
PRINT &B100
```

```
4
```

```
Ok
```

4-7:

Als antwoord op de gestelde vragen volgt hier een copie van het beeldscherm, zoals dat er uit zag na het invoeren van de gegeven uitdrukkingen:

```
PRINT 28
```

```
28
```

```
Ok
```

```
PRINT HEX$(10)
```

```
A
```

```
Ok
```

```
PRINT STR$(10)
```

```
10
```

```
Ok
```

```
PRINT VAL(STR$(10))
```

```
10
```

```
Ok
```

```
PRINT BIN$(8)
```

```
1000
```

```
Ok
```

4-8:

DEFINT A-Z declareert alle variabelen die met een van de letters A tot en met Z beginnen tot integer variabelen, dus ook variabele F is een integer variabele.

4-9:

Met het statement DEF SNG kunnen variabelen als variabele voor getallen met enkelvoudige nauwkeurigheid worden gedeclareerd.

4-10:

Zoals uit de volgende beeldschermweergave blijkt, geeft het programma bij uitvoering een foutboodschap. Na wijziging van regel 10 (de komma tussen de A en de D is gewijzigd in een koppelteken) loopt het programma goed.

```

10 DEFSTR A,D
20 A="tekst 1"
30 B="tekst 2"
40 PRINT A;B
50 END
Ok
run
Type mismatch in 30
Ok

```

```

10 DEFSTR A-D
20 A="tekst 1"
30 B="tekst 2"
40 PRINT A;B
50 END
Ok
run
tekst 1tekst 2
Ok

```

4-11:

Na het verwijderen van regel 10 geeft het programma bij uitvoering een foutboodschap. Zowel de oude situatie, als het gewijzigde programma zijn te zien in de volgende beeldschermweergave:

```

20 A="tekst 1"
30 B="tekst 2"
40 PRINT A;B
50 END
Ok
run
Type mismatch in 20
Ok

```

```

20 A$="tekst 1"
30 B$="tekst 2"
40 PRINT A$;B$
50 END
Ok
run
tekst 1tekst 2
Ok

```

5 Het toekennen van waarden aan variabelen

5-1

Met het REM-statement kan een programma van commentaar worden voorzien. Dit commentaar kan de leesbaarheid (begrijpbaarheid) van het programma verhogen.

5-2:

Deze regel is niet goed, omdat alle statements, die na het REM-statement in dezelfde programmaregel volgen, niet zullen worden uitgevoerd. Om het toekenningsstatement uit te laten voeren zal de regel er als volgt uit moeten zien:

```
10 LET A=10: REM waardetoekenning
```

5-3:

De listing van het gevraagde programma ziet er als volgt uit. Om te controleren of het programma inderdaad goed werkt, is na het uitvoeren van het programma het commando PRINT Z gegeven.

```
10 LET X=23456
20 LET Y=65432
30 LET Z=X+Y
40 END
RUN
Ok
PRINT Z
88888
Ok
```

5-4:

De variabelen uit het vorige programma zijn niet door ons gedeclareerd en daardoor automatisch variabelen voor getallen van dubbelvoudige nauwkeurigheid. De volgende beeldschermafdruk laat zien hoe we kunnen zorgen dat de variabelen voor integer getallen geschikt worden gemaakt.

```
5 DEFINT X-Z
10 LET X=23456
20 LET Y=65432!
30 LET Z=X+Y
40 END
Ok
RUN
Overflow in 20
Ok
```

```
5 DEFINT X-Z
10 LET X=2345
20 LET Y=5432
30 LET Z=X+Y
40 END
Ok
RUN
Ok
PRINT Z
7777
Ok
```

5-5:

De uitkomsten (door de computer berekend) zijn:

```
PRINT 8+3-6*2
-1
Ok
PRINT (8+3-6)*2
10
Ok
PRINT 2*2^3
16
Ok
PRINT (2*2)^3
64
Ok
```

5-6

Hier volgt het gevraagde programma, gevolgd door het resultaat zoals dat op het beeldscherm is te zien.

```
10 INPUT G
20 LET G=G^2
30 PRINT G
40 END
Ok
RUN
? 25
625
```

Dezelfde opdracht kan ook korter worden uitgevoerd:

```
10 INPUT G
20 PRINT G^2
30 END
Ok
RUN
? 12
144
Ok
```

5-7:

Doordat de cijfers 123 tussen aanhalingstekens staan, zijn deze cijfers een alfanumerieke constante. De variabele A is echter een numerieke variabele. Aan een numerieke variabele kan geen alfanumerieke waarde worden toegekend. Er zijn twee oplossingen:

- a) - Maak van variabele A een string-variabele (A\$).
- b) - Haal de aanhalingstekens rond de cijfers 123 weg.

```
5-8: 100 A$="ABR"  
      110 B$="ACADA",  
      120 C$="BRA"  
      130 P$=A$+B$+C$  
      140 PRINT P$  
      150 END  
      RUN  
      ABRACADABRA  
      Ok
```

```
5-9:  PRINT BIN$(32)  
      100000  
      Ok  
      PRINT VAL(BIN$(32))  
      100000  
      Ok  
      PRINT VAL("&B"+BIN$(32))  
      32  
      Ok
```

```
5-10: 10 INPUT "getal";G  
      20 PRINT G;"keer";G;"=";G*G  
      30 END  
      Ok  
      run  
      getal? 12  
      12 keer 12 = 144  
      Ok
```

5-11

Eerst volgt hier de listing van het programma, met een voorbeeld van de uitvoering:

```
10 INPUT "lengte,breedte,hoogte";L,B,H
20 PRINT "inhoud =";L*B*H
30 END
Ok
run
lengte,breedte,hoogte? 25,20,5
inhoud = 2500
Ok
```

Hetzelfde programma nogmaals, doch nu wordt er een foutje gemaakt bij het invoeren van de getallen:

```
10 INPUT "lengte,breedte,hoogte";L,B,H
20 PRINT "inhoud =";L*B*H
30 END
Ok
run
lengte,breedte,hoogte? 10,20
?? 5
inhoud = 1000
Ok
```

5-12:

```
10 p$=input$(5)
20 end
run
Ok
print p$
abcde
Ok
```

5-13:

```
10 INPUT "Waarde voor X";X
20 INPUT "Waarde voor Y";Y
30 PRINT "X =";X;"en Y =";Y
40 LET Z=X
50 LET X=Y
60 LET Y=Z
70 PRINT "X =";X;"en Y =";Y
80 END
Ok
run
Waarde voor X? 12
Waarde voor Y? 78
X = 12 en Y = 78
X = 78 en Y = 12
Ok
```


6 Afdrukken van gegevens

6-1:

- a) Tekstmode (SCREEN 0 en 1)
- b) Grafische mode (SCREEN 2 en 3)

6-2:

Aantal regels is 24.
Aantal tekens per regel is 32.

6-3:

Horizontaal 256 pixels, verticaal 192 pixels.

```
6-4:      100 SCREEN 3
          110 OPEN "GRP:" AS #1
          120 DRAW "BM80,50"
          130 PRINT #1,"MSX"
          140 DRAW "BM48,100"
          150 PRINT #1,"BASIC"
          160 GOTO 160
```

6-5:

1. CLS
2. PRINT CHR\$(12)
3. SHIFT-toets + HOME-toets.
4. CTRL-toets + letter L.

6-6:

Met het statement WIDTH

6-7:

De regellengte (default-waarde) is:

```
SCREEN 0 - 37 tekens.
SCREEN 1 - 29 tekens.
```

6-8:

De maximale regellengte is:

```
SCREEN 0 - 40 tekens.
SCREEN 1 - 32 tekens.
```

```

6-9:      100 SCREEN 0
          110 WIDTH 40
          120 INPUT "Naam           ";N$
          130 INPUT "Adres          ";A$
          140 INPUT "Woonplaats    ";W$
          150 PRINT:PRINT
          160 PRINT N$
          170 PRINT A$
          180 PRINT W$
          190 END

```

6-10:
 De functie TAB(X) kan worden vergeleken met de TAB-toets op de schrijfmachine, waarmee tabulatorstoppen kunnen worden ingesteld. De functie TAB wordt alleen gebruikt in combinatie met het PRINT of LPRINT-statement.

6-11:
 De waarde van de meest linker positie van een regel is 0. Het argument X mag een van de waarden 0 tot en met 255 hebben.

```

6-12:     100 SCREEN 0
          110 WIDTH 40
          120 INPUT "Naam           ";N$
          130 INPUT "Adres          ";A$
          140 INPUT "Woonplaats    ";W$
          150 PRINT:PRINT
          160 PRINT TAB(6);N$
          170 PRINT TAB(6);A$
          180 PRINT TAB(6);W$
          190 END

```

6-13:
 X - Horizontale positie (kolom) 0-39.
 Y - Verticale positie (regelnummer) 0-23.
 Z - 0 = cursor onzichtbaar.
 1 = cursor zichtbaar.

6-14:
 LOCATE ,,0

```

6-15:     100 CLS
          105 KEY OFF
          110 A$="MSX-BASIC"
          120 X=17

```

```

125 LOCATE 10,8
130 PRINT STRING$(19,"*")
135 LOCATE 10,9
140 PRINT "*" ;SPC(17);"*"
145 LOCATE 10,10
150 PRINT "*" ;SPC(4);A$;SPC(4);"*"
155 LOCATE 10,11
160 PRINT "*" ;SPACE$(X);"*"
165 LOCATE 10,12
170 PRINT STRING$(19,42)
180 END

```

6-16:

Met de functie SPC(X) kunnen we tussen twee af te drukken teksten een aantal spaties zetten. De functie wordt altijd gebruikt in combinatie met het PRINT of LPRINT-statement.

```

6-17:      100 WIDTH 40
           105 CLS
           110 LOCATE 7,0
           120 PRINT "A";SPC(7);"B";SPC(7);"C";SPC(7);"D"
           130 LOCATE 7,22
           140 PRINT "A";SPC(7);"B";SPC(7);"C";SPC(7);"D"
           150 GOTO 150

```

```

6-18:      100 REM * NU MET SPACE$ *
           110 WIDTH 40
           120 CLS
           130 I=7
           140 LOCATE 7,0
           150 PRINT "A";SPACE$(I);"B";SPACE$(I);"C";SPACE$(I);"D"
           160 LOCATE 7,22
           170 PRINT "A";SPACE$(I);"B";SPACE$(I);"C";SPACE$(I);"D"
           180 GOTO 180

```

6-19:

1. ASCII-codetabel.
2. MSX-codetabel.

6-20:

De code voor de letter P is 80.

6-21:
PRINT CHR\$(80)

6-22:

A = 65	41	J = 74	4A	S = 83	53
B = 66	42	K = 75	4B	T = 84	54
C = 67	43	L = 76	4C	U = 85	55
D = 68	44	M = 77	4D	V = 86	56
E = 69	45	N = 78	4E	W = 87	57
F = 70	46	O = 79	4F	X = 88	58
G = 71	47	P = 80	50	Y = 89	59
H = 72	48	Q = 81	51	Z = 90	5A
I = 73	49	R = 82	52		
0 = 48	30	5 = 53	35		
1 = 49	31	6 = 54	36		
2 = 50	32	7 = 55	37		
3 = 51	33	8 = 56	38		
4 = 52	34	9 = 57	39		

6-23:
PRINT CHR\$(75)+CHR\$(65)+CHR\$(82)+CHR\$(69)+CHR\$(76)

6-24:
Met het PRINT USING statement kunnen getallen en teksten in een bepaalde vorm en op een bepaalde positie (met behulp van een PRINT-masker) worden afgedrukt.

6-25:
"####"

6-26:
100 CLS
110 I=50
120 PRINT USING "####";I
130 I=I+100
140 IF I<1251 GOTO 120
150 PRINT
160 END

6-27:
"####.##"

```
6-28:    127.10
          24.00
          32.63
          114.91
          $2014.60
```

6-29:
Hiervoor moet men voor het PRINT-masker een plusteken plaatsen. Bijvoorbeeld: "+###.##"

```
6-30:    100 CLS
          110 I=-110
          120 PRINT USING "+###.##";I
          130 I=I+15
          140 IF I<116 GOTO 120
          150 PRINT
          160 END
```

7 Sprongen

7-1:

1. Onvoorwaardelijke sprongopdrachten.
2. Voorwaardelijke sprongopdrachten.

Bij een onvoorwaardelijke sprongopdracht wordt de sprong altijd uitgevoerd. Bij een voorwaardelijke sprongopdracht wordt de sprong alleen uitgevoerd wanneer er aan een bepaalde voorwaarde is voldaan.

7-2:

Undefined line number in 90

7-3:

IF <expressie> GOTO <regelnummer>

7-4:

Kleiner dan	<
Groter dan	>
Gelijk aan	=
Kleiner dan of gelijk aan	<= of =<
Groter dan of gelijk aan	>= of =>
Ongelijk aan	<> of ><

```
7-5:
IF F<=45 GOTO 120
```

```
7-6:      10 CLS
          20 PRINT "FAHRENHEIT    CELCIUS"
          25 PRINT
          30 F=32
          40 C=5/9*(F-32)
          50 PRINT USING "###";F,
          60 PRINT TAB(13),
          70 PRINT USING "##.##";C
          80 F=F+1
          90 IF F<46 GOTO 40
          100 END
```

```
7-7:      10 CLS
          11 INPUT "Beginwaarde";F
          12 INPUT "Eindwaarde ";E
          13 PRINT
          20 PRINT "FAHRENHEIT    CELCIUS"
          25 PRINT
          30 PRINT
          40 C=5/9*(F-32)
          50 PRINT USING "###";F,
          60 PRINT TAB(13),
          70 PRINT USING "###.##";C
          80 F=F+1
          90 IF F<E+1 GOTO 40
          100 END
```

7-8:
Een menu is een overzichtslijst van de routines die een bepaald programma kan uitvoeren. Uit de lijks kan steeds maar 1 routine tegelijkertijd worden gekozen.

```
7-9:
ON <variabele> GOTO R1,R2,...,Ri   of
ON <expressie> GOTO R1,R2,...,Ri
```

```
7-10:    100 WIDTH 40
          110 KEY OFF
          120 CLS
          130 LOCATE 8,1
          140 PRINT "OPPERVLAKTE BEREKENINGEN"
          150 LOCATE 8,4
          160 PRINT "1. Oppervlakte rechthoek"
          170 LOCATE 8,6
```

```

180 PRINT "2. Oppervlakte cirkel"
190 LOCATE 8,8
200 PRINT "3. Oppervlakte driehoek"
210 LOCATE 8,10
220 PRINT "4. Programma beëindiging"

```

```

7-11: 230 LOCATE 0,23
      240 INPUT "Uw keuze";K
      250 IF K<1 OR K>4 THEN PRINT "Verkeerde keuze
      1":GOTO 230
      260 IF K=4 THEN END
      270 ON K GOTO 300,450,600

```

```

7-12: 280 A$=INKEY$:IF A$="" GOTO 280
      290 GOTO 120
      300 CLS
      310 INPUT "Rechthoekszijde a";A
      320 INPUT "Rechthoekszijde b";B
      330 PRINT
      340 O=A*B
      350 PRINT "Oppervlakte rechthoek is ";O
      360 GOTO 280
      450 CLS
      460 INPUT "Straal r";R
      470 PRINT
      480 O=3.14*R^2
      490 PRINT "Oppervlakte cirkel is ";O
      500 GOTO 280
      600 CLS
      610 INPUT "Basis b";B
      620 INPUT "Hoogte h";H
      630 PRINT
      640 O=.5*B*H
      650 PRINT "Oppervlakte driehoek is ";O
      660 GOTO 280

```

7-13:
 Het komt nogal een voor dat in een programma op verschillende plaatsen steeds dezelfde rij statements wordt gebruikt. In dat geval is het efficiënter om van die rij statements een op zichzelf staan programmadeel te maken. Dat programmadeel kan dan, indien nodig, vanuit het hoofdprogramma worden aangeroepen.

7-14:

1. GOSUB
2. ON...GOSUB

7-15:

ON <variabele> GOSUB R1,R2,...,Ri

7-16:

RETURN

7-17:

```
100 WIDTH 40
110 KEY OFF
120 CLS
130 LOCATE 8,1
140 PRINT "OPPERVLAKTE BEREKENINGEN"
150 LOCATE 8,4
160 PRINT "1. Oppervlakte rechthoek"
170 LOCATE 8,6
180 PRINT "2. Oppervlakte cirkel"
190 LOCATE 8,8
200 PRINT "3. Oppervlakte driehoek"
210 LOCATE 8,10
220 PRINT "4. Programma beeindiging"
230 LOCATE 0,23
240 INPUT "Uw keuze";K
250 IF K<1 OR K>4 THEN PRINT "Verkeerde keuze
!" :GOTO 230
260 IF K=4 THEN END
270 ON K GOSUB 300,450,600
280 A$=INKEY$:IF A$="" GOTO 280
290 GOTO 120
300 CLS
310 INPUT "Rechthoekszijde a";A
320 INPUT "Rechthoekszijde b";B
330 PRINT
340 O=A*B
350 PRINT "Oppervlakte rechthoek is ";O
360 RETURN
450 CLS
460 INPUT "Straal r";R
470 PRINT
480 O=3.14*R^2
490 PRINT "Oppervlakte cirkel is ";O
500 RETURN
600 CLS
610 INPUT "Basis b";B
620 INPUT "Hoogte h";H
```

```

630 PRINT
640 O=.5*B*H
650 PRINT "Oppervlakte driehoek is ";O
660 RETURN

```

8 Lussen en opslaan

8-1:
De logische operators zijn: AND, OR en NOT.

8-2:

X	Y	X OR Y
niet waar	niet waar	niet waar
niet waar	waar	waar
waar	niet waar	waar
waar	waar	waar

8-3:

```
IF X=2 AND Y<6 THEN A=A+4
```

8-4:

```
IF X=3 OR Y>4 THEN A=A+2 ELSE A=A+4
```

8-5:

```

100 CLS
110 INPUT "Eerste getal";G
120 P=1
130 FOR I=G TO G+4
140 P=P*I
150 NEXT I
160 PRINT
170 PRINT "Het produkt is ";P
180 PRINT
190 END

```

8-6:

```

100 CLS
110 INPUT "Eerste getal";G
115 INPUT "Laatste getal";L
120 P=1
130 FOR I=G TO L
140 P=P*I

```

```
150 NEXT I
160 PRINT
170 PRINT "Het produkt is ";P
180 PRINT
190 END
```

```
8-7:    100 CLS
        110 INPUT "Eerste getal ";E
        120 INPUT "Laatste getal";L
        130 S=0
        140 FOR I=E TO L STEP 2
        150 S=S+I
        160 NEXT I
        170 PRINT
        180 PRINT "De som is ";S
        190 PRINT
        200 END
```

```
8-8:    2
        -13
        37
```

```
8-9:    100 CLS
        110 INPUT "Laagste grens";X
        120 INPUT "Hoogste grens";Y
        125 PRINT
        130 PRINT "De priemgetallen zijn:"
        140 D=2
        150 IF X/D=INT(X/D) GOTO 190
        160 D=D+1
        170 IF D<=SQR(X) GOTO 150
        180 PRINT X;
        190 X=X+1
        200 IF X<=Y GOTO 140
        210 END
```

8-10:
Het commando CSAVE wordt gebruikt voor het opslaan van programma's vanuit het geheugen op cassette in binaire vorm. Dit in tegenstelling tot het commando SAVE, waarmee de programma's in ASCII-formaat op cassette worden geschreven.

```

8-11:      100 CLS
           110 INPUT "Variabele a";A
           120 INPUT "Variabele b";B
           130 INPUT "Variabele c";C
           140 PRINT
           150 X=(A*B-C)/2.5
           160 PRINT "x =";X
           170 PRINT
           180 END

8-12:      SAVE "CAS:BER1"
           Ok
           NEW
           Ok

8-13:      180 CLS
           190 INPUT "Variabele d";D
           200 INPUT "Variabele e";E
           210 PRINT
           220 Y=SQR(D+E)
           230 PRINT "y =";Y
           240 PRINT
           250 END

8-14:      SAVE "CAS:BER2"
           Ok
           NEW
           Ok

8-15:      250 Z=(X-Y)/2
           260 PRINT "z =";Z
           270 PRINT
           280 END

8-16:      Spoel eerst de cassette terug! Geef daarna:
           MERGE "CAS:BER1"

8-17:      MERGE "CAS:BER2"

```


8-18:

```
180
250
RUN
```

8-19:

```
100 CLS
110 PRINT "Stel recorder in op terug-"
120 PRINT "spoelen en druk hierna de"
130 PRINT "RETURN-toets in."
140 IF INKEY$<>CHR$(13) GOTO 140
150 MOTOR ON
160 PRINT
170 PRINT "Druk wanneer band volledig"
180 PRINT "is teruggespoeld de RETURN-"
190 PRINT "toets in."
200 IF INKEY$<>CHR$(13) GOTO 200
210 MOTOR OFF
220 PRINT
230 PRINT "Stel recorder in op afspelen"
240 PRINT "en druk daarna de RETURN-"
250 PRINT "toets in, waarna de band"
260 PRINT "naar het begin wordt ge-"
270 PRINT "transporteerd."
275 IF INKEY$<>CHR$(13) GOTO 275
280 MOTOR ON
290 FOR I=1 TO 5000:NEXT I
300 MOTOR OFF
310 PRINT
320 PRINT "Band heeft begin bereikt!"
330 END
```

9 Tijd en fouten

9-1:

```
10 SCREEN 0: WIDTH 40: CLS
20 ON INTERVAL=200 GOSUB 60
30 PRINT "Lees dit gauw, want over enkel
e secon- den wordt het beeldscherm gewi
st."
40 INTERVAL ON
50 GOTO 50
60 CLS
70 RETURN
```

9-2:

Na het uitvoeren van het INTERVAL OFF wordt niet meer gecontroleerd of er een tijdsinterval is verstreken. Na het uitvoeren van het statement INTERVAL STOP wordt nog wel gecontroleerd of er een tijdsinterval is verstreken, doch er wordt niet op dit verstrijken gereageerd. Pas nadat het statement INTERVAL ON is gegeven, zal op een eventueel verstreken tijdsinterval worden gereageerd.

9-3:

```
10 SCREEN 0: WIDTH 40: CLS
20 I$=""
30 INPUT "Af te drukken tekst";I$
40 IF LEN(I$)<40 THEN PRINT I$:GOTO 20
50 PRINT "Past niet op de regel"
60 GOTO 20
Ok
run
Af te drukken tekst? aaaaaaaaaa
aaaaaaaaaa
Af te drukken tekst? bbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbb
Past niet op de regel
Af te drukken tekst?
```

9-4:

```
10 SCREEN 0: WIDTH 40: CLS: VP=2
20 PRINT "0.....1.....2.....
3....."
30 I$=""
40 LOCATE 0,15
50 INPUT "Af te drukken tekst";I$
60 IF LEN(I$)>40 GOTO 30
70 HP=INT((40-LEN(I$))/2)
80 VP=VP+1
90 LOCATE HP,VP
100 PRINT I$
110 GOTO 30
```

Dit programma geeft het volgende resultaat op het beeldscherm:

0.....1.....2.....3.....

Het gaat natuurlijk
om de formule
in regel 70
De rest is alleen
maar
franje

```
9-5: 10 a$="abcde"
      20 a$=right$(a$,4)+left$(a$,1)
      30 print a$
      40 end
      run
      bcdea
      Ok

9-6: 10 input "tekst";a$
      20 l=len(a$)
      30 a$=right$(a$,l-1)+left$(a$,1)
      40 print a$
      run
      tekst? wessel
      esselw

9-7: 10 INPUT "tekst";A$
      20 L=LEN(A$)
      30 LOCATE 0,10
      40 A$=RIGHT$(A$,L-1)+LEFT$(A$,1)
      50 PRINT A$
      60 FOR I=1 TO 100: NEXT I
      70 GOTO 30

9-8: 5 SCREEN 0: WIDTH 40: CLS
      10 INPUT "tekst";A$
      15 CLS
      20 L=LEN(A$)
      30 LOCATE INT((40-L)/2),10
      40 A$=RIGHT$(A$,L-1)+LEFT$(A$,1)
      50 PRINT A$
      60 FOR I=1 TO 100: NEXT I
      70 GOTO 30
```

```

9-9:      10 SCREEN 0: WIDTH 40: CLS
          20 INPUT "Tekst";T$
          30 INPUT "Te tellen letter";L$
          40 L=LEN(T$): T=0: CLS
          50 FOR I=1 TO L
          60 IF MID$(T$,I,1)=L$ THEN T=T+1
          70 NEXT I
          80 PRINT "De ";L$;" komt";T;"keer voor i
          n:"
          90 PRINT T$
          100 END

```

Het resultaat van dit programma ziet er als volgt uit:

```

Tekst? het msx basic leerboek heeft mij
tot op heden al heel wat statements gele
erd.
Te tellen letter? a
De a komt 4 keer voor in:
het msx basic leerboek heeft mij tot op
heden al heel wat statements geleerd.

```

9-10:
Dit programma geeft de foutboodschap "Syntax error", die blijkbaar wordt veroorzaakt door regelnummer 30.

```

9-11:      10 ON ERROR GOTO 100
          20 A=1:B=2
          30 A+B
          40 PRINT A;B
          50 GOTO 30
          100 BEEP: BEEP: BEEP
          110 PRINT "Regel";ERL;"heeft foutcode";E
          RR
          120 PRINT "de regel luidt:"
          130 LIST.
          140 END
          Ok
          run
          Regel 30 heeft foutcode 2
          de regel luidt:
          30 A+B
          Ok

```


9-12:

Beweeg de cursor met behulp van de pijltjes-toetsen (cursor control toetsen) naar regelnummer 30. Zet de cursor op de "A". Druk op de INS-toets en geef nu in:

A=

Door nu op de RETURN-toets te drukken is regelnummer 30 gewijzigd in:

30 A=A+B

Bij het opnieuw uitvoeren van het programma zal blijken dat er geen fout meer optreedt.

10 Het programmeren van functietoetsen

10-1:

Wanneer we functietoets F5 indrukken, wordt op het scherm "run" afgedrukt en vervolgens het in het geheugen staande programma uitgevoerd.

10-2:

KEY LIST

10-3:

KEY OFF

10-4:

d) Wanneer functietoets F10 wordt ingedrukt, wordt voordat het programma wordt uitgevoerd eerst het beeldscherm schoongemaakt. Dit is niet het geval bij het indrukken van functietoets F5.

10-5:

KEY <functietoetsnummer>,<string-uitdrukking>

10-6:

KEY 2,"KEY OFF"+CHR\$(13)+"CLS"+CHR\$(13)

10-7:

KEY 3,"KEY ON"+CHR\$(13)+"CLS"+CHR\$(13)

10-8:

ON KEY GOSUB <lijst met regelnummers>

10-9: KEY(funcnietoets) ON

```
10-10:       100 CLS
          110 KEY 4,"cirkel"
          120 KEY(4) ON
          130 ON KEY GOSUB ,,,250
          140 GOTO 140
          250 CLS
          260 PRINT "BEREKENING CIRKEL"
          270 PRINT
          280 INPUT "Straal r";R
          290 PRINT
          300 O=3.14*R^2
          310 PRINT "Inhoud cirkel is";O
          320 RETURN
```

10-11: KEY <functienummer>,""

10-12:

Wanneer het in een bepaald gedeelte van het programma niet wenselijk is dat dit wordt onderbroken, maar dat het indrukken van een functietoets wel moet worden onthouden, dan kan dit worden geregeld met het statement:

KEY(funcnietoets) STOP.

10-13:

Een programma kan tijdelijk worden gestopt door het indrukken van de STOP-toets.

10-14:

Een programma wordt beëindigd door het tegelijkertijd indrukken van de CTRL-toets en de STOP-toets.

11 Werken met vaste gegevens

11-1:

Met het statement READ kunnen we alfanumerieke- en numerieke constanten lezen van een DATA-regel.

11-2:

Er mogen in een en dezelfde DATA-regel zowel numerieke- als alfanumerieke constanten staan.

11-3:

- a) Wanneer na de DATA-regel nog een DATA-regel volgt, zal het READ-statement het eerste element van deze regel lezen.
- b) Wanneer na de DATA-regel geen DATA-regel meer volgt, zal de computer de volgende foutboodschap afdrucken: "Out of DATA in".

11-4:

Met het statement RESTORE.

```
11-5:      100 CLS
           110 S=0
           120 FOR I=1 TO 8
           130 READ G
           140 S=S+G
           150 NEXT I
           160 PRINT "Het rekenkundig gemiddelde is";S/8
           170 PRINT
           180 DATA 4,78,2.5,34,9,123,14,42
           190 END
```

11-6:

```
100 CLS
110 S=0
120 PRINT "INVOEREN VAN CIJFERS:"
130 PRINT
140 FOR I=1 TO 10
150 READ N$
160 PRINT N$;TAB(10);
170 INPUT C
180 S=S+C
190 NEXT I
200 PRINT
210 PRINT "Gemiddelde cijfer is";S/10
```

```

220 PRINT
230 INPUT "Volgende berekening (j/n)";J$
240 IF J$="J" OR J$="j" THEN RESTORE:GOTO 100
250 DATA Robert,Peter,Marcel,Frank,Petra,Anne
tte,Margriet,Paul,Albrecht,Ellen
260 END

```

12 Opslaan van gegevens

12-1:

Vier velden; 0, 1, 2 en 3.

```

12-2:      10 SCREEN 0: WIDTH 40: CLS
           20 DIM A$(3)
           30 FOR I=0 TO 3
           40 READ A$(I)
           50 NEXT I
           60 END
           70 DATA appel,peer,pruim,kers

```

```

12-3:      10 SCREEN 0: WIDTH 40: CLS
           20 DIM A$(3)
           30 FOR I=0 TO 3
           40 INPUT "item";A$(I)
           50 NEXT I
           60 END

```

```

12-4:      10 SCREEN 0: WIDTH 40: CLS
           20 DIM A$(3)
           30 FOR I=0 TO 3
           40 INPUT "item";A$(I)
           50 NEXT I
           60 FOR I=0 TO 3
           70 PRINT A$(I)
           80 NEXT I
           90 END

```

12-5:

Met FRE(0) wordt de totale, voor BASIC beschikbare, geheugenruimte opgevraagd.

12-6:
CLEAR 500

12-7:
Met FRE("") wordt de geheugenruimte die nog beschikbaar is voor het opslaan van alfanumerieke gegevens (strings) opgevraagd.

12-8:
In het geheugengebied dat met het statement CLEAR voor het opslaan van strings is gereserveerd. Na aanschakelen van de computer is dit geheugengebied 200 bytes groot.

```
12-9:      10 SCREEN 0: WIDTH 40: CLS
           20 DIM A$(10)
           30 FOR I=0 TO 10
           40 PRINT "vrije geheugenruimte =";FRE(0)
           50 PRINT "vrije string-ruimte =";FRE("")
           60 INPUT "item";A$(I)
           70 NEXT I
           80 END
```

```
12-10:     10 SCREEN 0: WIDTH 40: CLS
           20 DIM A$(9,2)
           30 FOR I=0 TO 9
           40 CLS
           50 FOR J=0 TO 2
           60 PRINT "item";I;J;
           70 INPUT A$(I,J)
           80 NEXT J
           90 NEXT I
           100 CLS
           110 INPUT "rij ";R
           120 INPUT "kolom";K
           130 PRINT "item";R;K;"=" ";A$(R,K)
           140 GOTO 110
```

```
12-11:     10 SCREEN 0: WIDTH 40: CLS
           20 DIM A$(9,2)
           30 FOR I=0 TO 9
           40 CLS
           50 FOR J=0 TO 2
           60 PRINT "item";I;J;
```

```

70 INPUT A$(I,J)
80 NEXT J
90 NEXT I
100 CLS
110 PRINT "Zoeken:"
120 INPUT "in welke kolom";K
130 INPUT "naar welke tekst";T$
140 FOR R=0 TO 9
150 IF INSTR(A$(R,K),T$)<>0 GOTO 180
160 NEXT R
170 PRINT T$;" komt niet voor"
175 GOTO 110
180 PRINT "rij =" ;R
190 PRINT "kolom=" ;K
200 PRINT A$(R,K)
210 PRINT
220 GOTO 110

```

12-12: Foutboodschap nummer 10, "Redimensioned array".

12-13: ERASE <variabelenaam>

```

12-14: 3000 CLS
3010 ERASE A$
3020 DIM A$(20)
3030 FOR I=0 TO 20
3040 PRINT "ITEM";I;"= ";
3050 INPUT A$(I)
3060 NEXT I
3070 RETURN

```

```

12-15: 4000 FOR J=0 TO 2
4010 CLS
4020 FOR I=0 TO 6
4030 PRINT A$(J*7+I)
4040 NEXT I
4050 PRINT
4060 PRINT "RETURN = doorgaan"
4070 IF INKEY$<>CHR$(13) GOTO 4070
4080 NEXT J
4090 RETURN

```

```

12-16: 5000 PRINT
5010 PRINT "RETURN = motor aan/uit"
5020 PRINT "ESC = band staat goed"
5030 M=0
5040 IF M=0 THEN MOTOR OFF ELSE MOTOR ON
5050 I$=INKEY$
5060 IF I$=CHR$(27) THEN MOTOR OFF: RETU
RN
5070 IF I$<>CHR$(13) GOTO 5040
5080 M=NOT(M)
5090 GOTO 5040

```

```

12-17: 1000 CLS
1010 PRINT "LEG EEN LEGE CASSETTE IN DE
RECORDER"
1020 GOSUB 5000: 'MOTOR AAN/UIT
1030 PRINT
1040 PRINT "ZET DE RECORDER IN DE STAND
OPNAME"
1050 PRINT
1060 INPUT "FILE-NAAM";N$
1070 N$="CAS:"+N$
1080 OPEN N$ FOR OUTPUT AS #1
1090 FOR I=0 TO 20
1100 PRINT #1,A$(I)
1110 NEXT I
1120 CLOSE #1
1130 RETURN

```

```

12-18: 2000 CLS
2010 PRINT "LEG DE CASSETTE MET HET TE L
EZEN BESTANDIN DE CASSETTERECORDER"
2020 PRINT "SPOEL DE CASSETTE TERUG NAAR
HET BEGIN VAN HET BESTAND"
2030 GOSUB 5000: 'MOTOR AAN/UIT
2040 PRINT
2050 PRINT "ZET DE RECORDER IN DE STAND
WEERGAVE"
2060 PRINT
2070 INPUT "FILE-NAAM";N$
2080 N$="CAS:"+N$
2090 OPEN N$ FOR INPUT AS #1
2100 FOR I=0 TO 20
2110 IF EOF(1)<>0 GOTO 2140
2120 LINE INPUT #1,A$(I)

```

```
2130 NEXT I
2140 CLOSE #1
2150 RETURN
```

```
12-19: 100 CLS
110 PRINT "1 = NIEUWE ARRAY MAKEN"
120 PRINT "2 = ARRAY NAAR CASSETTE SCHRI
JVEN"
130 PRINT "3 = ARRAY VAN CASSETTE LEZEN"
140 PRINT "4 = ARRAY OP BEELDSCHERM LIJS
TEN"
150 PRINT "5 = PROGRAMMABEEINDIGING"
160 PRINT
170 INPUT "KIES EEN FUNCTIE: ";F
180 IF F<1 OR F>5 GOTO 170
170 ON F GOSUB 3000,1000,2000,4000,6000
180 GOTO 100
```

```
12-20: 1 '***** INITIALISATIE *****
10 SCREEN 0
20 WIDTH 40
30 CLS
40 CLEAR 500
50 DIM A$(20)
99 '***** MENU *****
100 CLS
110 PRINT "1 = NIEUWE ARRAY MAKEN"
120 PRINT "2 = ARRAY NAAR CASSETTE SCHRI
JVEN"
130 PRINT "3 = ARRAY VAN CASSETTE LEZEN"
140 PRINT "4 = ARRAY OP BEELDSCHERM LIJS
TEN"
150 PRINT "5 = PROGRAMMABEEINDIGING"
160 PRINT
170 INPUT "KIES EEN FUNCTIE: ";F
180 IF F<1 OR F>5 GOTO 170
190 ON F GOSUB 3000,1000,2000,4000,6000
200 GOTO 100
999 '***** CASSETTE-OUTPUT *****
1000 CLS
1010 PRINT "LEG EEN LEGE CASSETTE IN DE
RECORDER"
1020 GOSUB 5000: 'motor aan/uit
1030 PRINT
1040 PRINT "ZET DE RECORDER IN DE STAND
```

```

OPNAME"
1050 PRINT
1060 INPUT "FILE-NAAM";N$
1070 N$="CAS:"+N$
1080 OPEN N$ FOR OUTPUT AS #1
1090 FOR I=0 TO 20
1100 PRINT #1,A$(I)
1110 NEXT I
1120 CLOSE #1
1130 RETURN
1999 '***** CASSETTE-INPUT *****
2000 CLS
2010 PRINT "LEG DE CASSETTE MET HET TE L
EZEN BESTANDIN DE CASSETTERECORDER"
2020 PRINT "SPOEL DE CASSETTE TERUG NAAR
HET BEGIN VAN HET BESTAND"
2030 GOSUB 5000: 'motor aan/uit
2040 PRINT
2050 PRINT "ZET DE RECORDER IN DE STAND
WEERGAVE"
2060 PRINT
2070 INPUT "FILE-NAAM";N$
2080 N$="CAS:"+N$
2090 OPEN N$ FOR INPUT AS #1
2100 FOR I=0 TO 20
2110 IF EOF(1)<>0 GOTO 2140
2120 LINE INPUT #1,A$(I)
2130 NEXT I
2140 CLOSE #1
2150 RETURN
2999 '***** ARRAY-MAKEN *****
3000 CLS
3010 ERASE A$
3020 DIM A$(20)
3030 FOR I=0 TO 20
3040 PRINT "ITEM";I;" = ";
3050 INPUT A$(I)
3060 NEXT I
3070 RETURN
3999 '***** ARRAY-LIJSTEN *****
4000 FOR J=0 TO 2
4010 CLS
4020 FOR I=0 TO 6
4030 PRINT A$(J*7+I)
4040 NEXT I
4050 PRINT
4060 PRINT "RETURN = doorgaan"

```



```

4070 IF INKEY$<>CHR$(13) GOTO 4070
4080 NEXT J
4090 RETURN
4999 '***** MOTOR AAN/UIT *****
5000 PRINT
5010 PRINT "RETURN = motor aan/uit"
5020 PRINT "ESC = band staat goed"
5030 M=0
5040 IF M=0 THEN MOTOR OFF ELSE MOTOR ON
5050 I$=INKEY$
5060 IF I$=CHR$(27) THEN MOTOR OFF: RETU
RN
5070 IF I$<>CHR$(13) GOTO 5040
5080 M=NOT(M)
5090 GOTO 5040
5999 '***** EINDE PROGRAMMA *****
6000 CLS
6010 PRINT "PROGRAMMA BEEINDIGD"
6020 END

```

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Akkermans, Wessel

MSX leerboek / Wessel Akkermans, Piet den Heijer. — Oosterend:
Stark-Texel

Opdrachten bij

DI. 1

ISBN 90 6398 596 7

SISO 365.3 UDC 681.3

Trefw.: MSX (computer)

uitgeverij STARK-TEXEL

postbus 302 - 1794 ZG Oosterend tel. 02223 - 661

