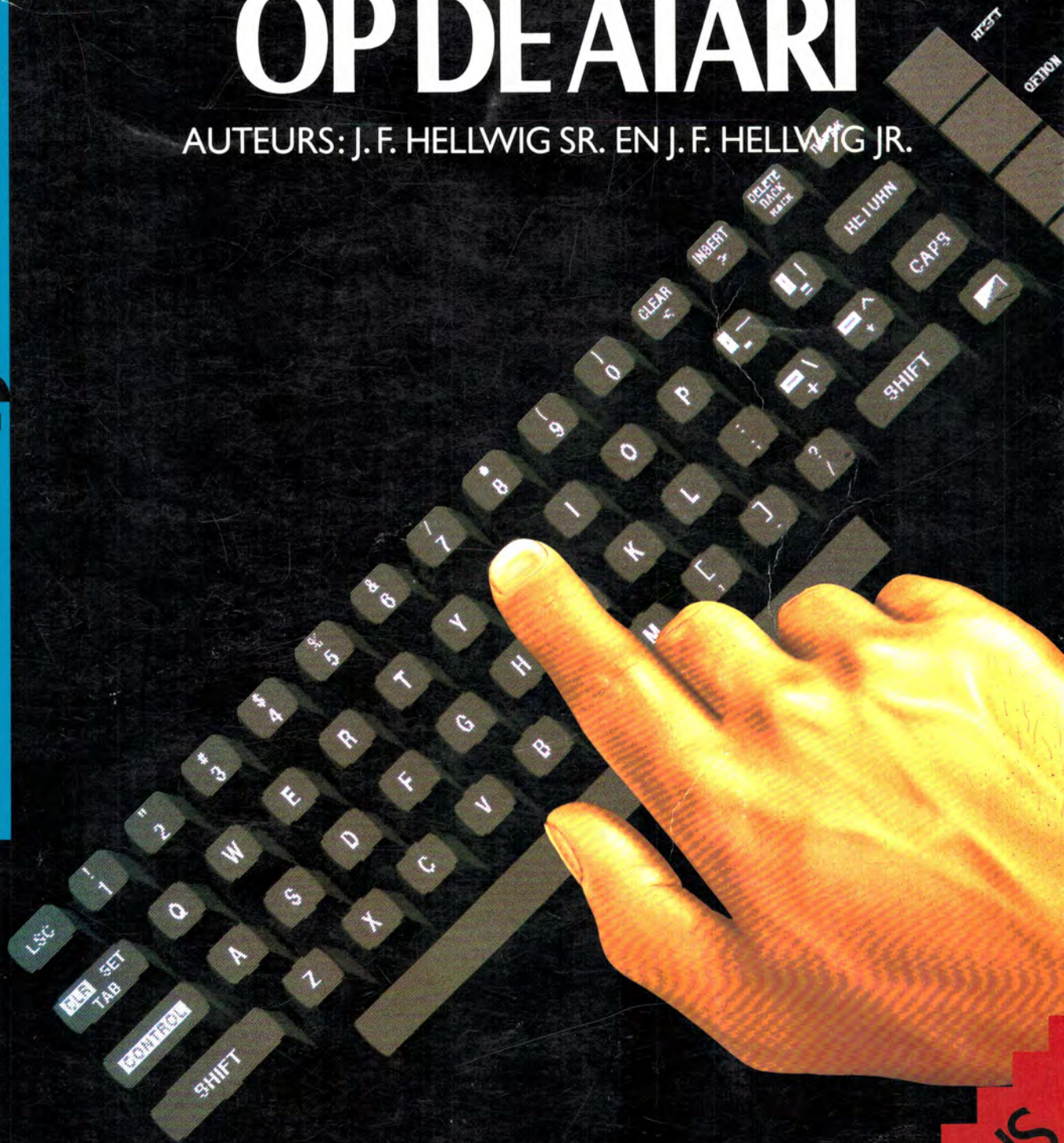


ATARI
MUSEUM.NL

LEER IN BASIC PROGRAMMEREN OP DE ATARI

AUTEURS: J. F. HELLWIG SR. EN J. F. HELLWIG JR.

Wolters Software



CURSUS



Leer in BASIC programmeren op de ATARI





J.F. Hellwig sr.

en

J.F. Hellwig jr.

**Leer
in BASIC programmeren
op de ATARI**

Wolters software



Copyright © 1985, Wolters Noordhoff bv

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen, of op enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Vormgeving: Sepp Bader, Goor

Boekproductie: Ar Nederhof, Amsterdam

ISBN 9001 97036 2

Inhoud

1	Inleiding 7
2	De kennismaking 9 Installeren, aansluiten en testen
3	Het toetsenbord 17 Speciale toetsen – grafische symbolen
4	Instructies 25 PRINT-instructie – variabelen – LET-instructie
5	Programma's en lussen 35 Opbouw programma's – GOTO-instructie
6	Het verbeteren van fouten en het SAVEN van programma's 43 Gebruik van de cassetterecorder
7	Condities 49 REM-instructie – leestekens – relatietekens
8	Rekenen en INPUT-instructie 57 Bewerkingsvolgorde
9	Geneste lussen en kleur 63 FOR...NEXT-instructie – GRAPHICS 0 – SETCOLOR-instructie
10	DATA- en READ-instructies 73 DATA-regels – vlag plaatsen – RESTORE-instructie
11	Geluid 79 Geluidsregisters – SOUND-instructie
12	POSITION en beweging 87 POSITION-instructie – bewegende beelden – afkortingen
	Getallen en functies 93 E-notatie – rekenkundige functies – numerieke functies

13

POKE en PEEK ⁹⁹
ROM-geheugen – POKE-instructie – PEEK-functie

14

Array's 107
DIM-instructie →matrix

15

String-functies 113
ASC-functie – CHR\$-functie – LEN-functie

16

Programma's 119
Ontwerpen – GOSUB-instructie – supervisor

17

GRAPHICS 0, 1 en 2 127
Grote letters – grafische symbolen

18

GRAPHICS 3, 5 en 7 135
PLOT-instructie – DRAWTO-instructie – COLOR-instructie

19

GRAPHICS 4, 6 en 8 143
Cirkel en elips

20

GRAPHICS 9, 10 en 11 149
Kleurmogelijkheden

A

Oplossingen van de opdrachten 155

B

Foutboodschappen 165

C

BASIC-commando's, instructies en functies 167

D

De ATASCII-code – de CHR\$-code – de schermcode 171

E

Belangrijke POKE- en PEEK-adressen 177

F

Geluid 181

Register 185

Inleiding

Het doel van dit boek is u informatie te geven over het installeren en het gebruik van een van de meest complete home-computers, de ATARI 800XL. We hebben het steeds over de ATARI 800XL als aanduiding voor alle ATARI 8 bits computers: 600XL, 800XL, 65XE en 130XE.

De computer kan worden gezien als een hulpmiddel, een instrument om allerlei dingen voor ons uit te voeren. Hoe ingewikkeld de computer ook in elkaar mag zitten, het leren omgaan met de computer is eenvoudiger dan de meesten denken.


U hoeft voor het gebruik van de ATARI 800XL geen wiskundeknobbel te hebben of een kei te zijn in de Engelse taal. De paar Engelse termen die bij het gebruik van de computer onvermijdelijk zijn, zijn over het algemeen woordjes van één lettergreep die, indien nodig, ook nog in dit boek worden verklaard.

De computer werkt naar aanleiding van programma's, lijsten met instructies samengesteld door een programmeur, die de computer stap voor stap vertellen wat er moet worden uitgevoerd. De programma's zijn geschreven in een taal die de computer begrijpt, de programmeertaal.

De programmeertaal die is ingebouwd in de ATARI 800XL is BASIC (Beginners All-purpose Symbolic Instruction Code). Deze taal maakt het voor iedereen die geen ervaring met het programmeren heeft gemakkelijk om in korte tijd met de computer te leren omgaan.

In ieder hoofdstuk van dit boek vindt u taken waarmee u zelf de vorderingen die u maakt kunt controleren. Een overzicht van de oplossingen voor deze taken vindt u in aanhangsel A. Tevens zijn alle hoofdstukken voorzien van een programma op cassettebandje. In de programma's op deze bandjes worden de belangrijkste punten uit de hoofdstukken nogmaals behandeld.

Dit boek bestaat uit twee delen. De behandelde stof uit het eerste deel (hoofdstuk 1 tot en met 10) stelt u in



staat verschillende soorten eenvoudige programma's zelf te maken, zoals programma's voor berekeningen, spelletjes, eenvoudige tekeningen en muziek.

De stof die in het tweede deel wordt behandeld (vanaf hoofdstuk 11) gaat uitgebreider in op de mogelijkheden die de computer biedt, zoals de functies, het opstellen van ingewikkelder programma's, het gebruik van reeksen en de grote grafische mogelijkheden van de ATARI 800XL.

De kennismaking

Dit hoofdstuk gaat over het installeren van de ATARI 800XL en de cassetterecorder, het testen van de computer en het laden van programma's vanaf een cassettebandje.

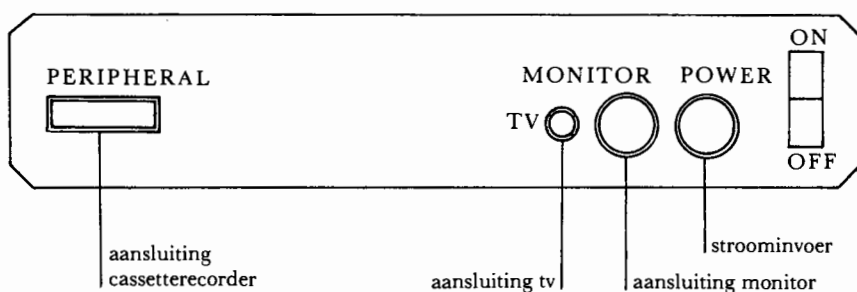
1.1 Het aansluiten van de computer op de tv en op de cassetterecorder

Plaats de ATARI 800XL, de cassetterecorder en de televisie (of monitor) op een grote tafel. Het prettigst werkt u indien de tv ongeveer vijftien centimeter hoger staat dan de computer.

U heeft een verlengsnoer met minstens drie stopcontacten nodig, omdat het basissysteem – computer, tv en recorder – moet worden aangesloten op de netspanning.

Het is verstandig zoveel mogelijk ruimte te laten tussen de tv en de rest van het systeem. In de eerste plaats voor uw eigen comfort, maar ook om de mogelijkheid van elektromagnetische interferentie (EMI) door de televisie zo klein mogelijk te maken.

De computer is aan de rechterzijde en aan de achterkant voorzien van verschillende aansluitmogelijkheden. De aansluitingen aan de rechterzijde zijn voor joy-sticks die in dit boek niet worden behandeld. De aansluitingen aan de achterzijde worden op onderstaande tekening weergegeven.



Voordat u een apparaat aansluit moet u eerst de stand van de hoofdschakelaar controleren. Deze moet in de stand 'OFF' staan.

Steek het ene einde van de RF-kabel (1) in de met TV gemerkte ingang aan de achterzijde van de computer en steek de andere zijde in de antenne-ingang van uw televisietoestel.

Steek de ronde 7-pins DIN-stekker van de adapterkabel (2) in de met POWER aangegeven ingang aan de achterzijde van de computer en steek de stekker aan de andere zijde in het stopcontact.

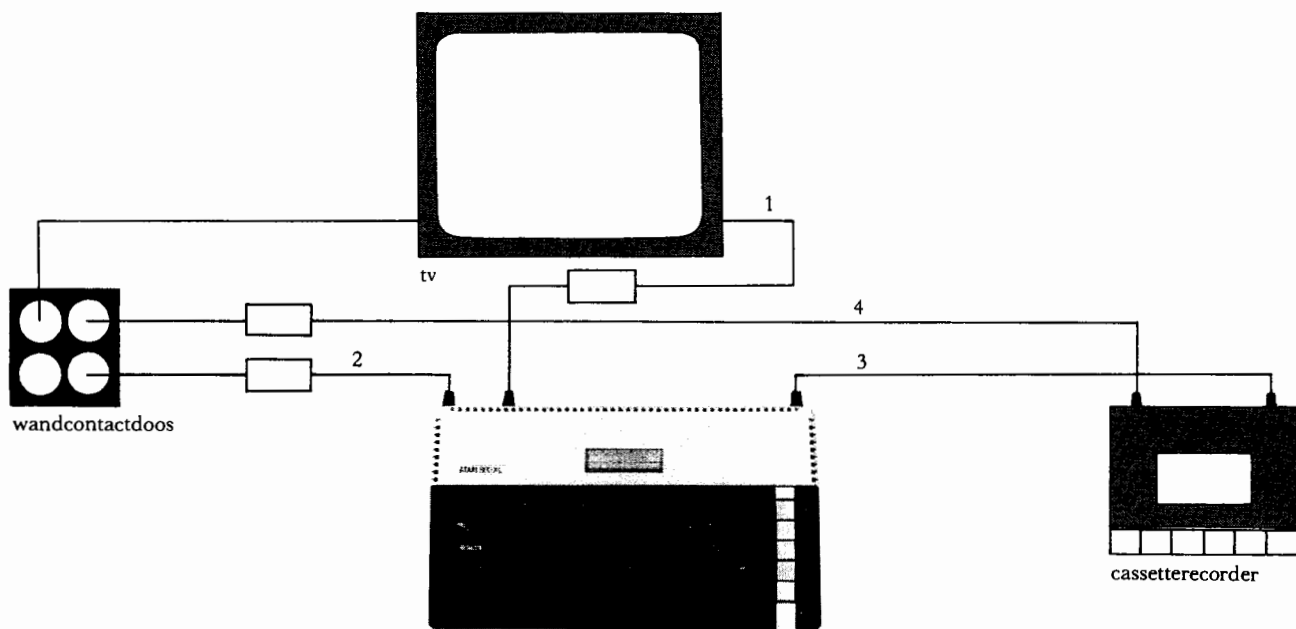
Steek een uiteinde van het recorder-datasnoer (3) in de met PERIPHERAL gemerkte aansluiting aan de achterzijde van de computer en steek de andere zijde in de I/O-aansluiting aan de achterzijde van de recorder.

Steek de ronde stekker van de recorder-netadapter (4) in de met POWER gemerkte ingang van de cassetterecorder en steek de stekker aan de andere zijde in het stopcontact.

Plaats de recorder (indien deze in gebruik is) zover mogelijk van de tv

vandaan om beschadigingen aan het cassettebandje door elektromagnetische interferentie te voorkomen.

Het gehele systeem is nu aangesloten zoals op onderstaande tekening is aangegeven.



1.2 Het inschakelen van de computer

Controleer of de 220 volt spanning op de juiste wijze is aangesloten en schakel dan de tv en de computer in, door de hoofdschakelaar in de stand 'ON' te plaatsen. Het rode waarschuwingslampje aan de rechterzijde boven het woord POWER moet nu branden.

Indien de computer en de tv lange tijd aanstaan zonder dat de computer wordt gebruikt, zal de kleur van het beeld veranderen.

Dit is een veiligheid die in de computer is ingebouwd om beschadiging van de beeldbuis te voorkomen.

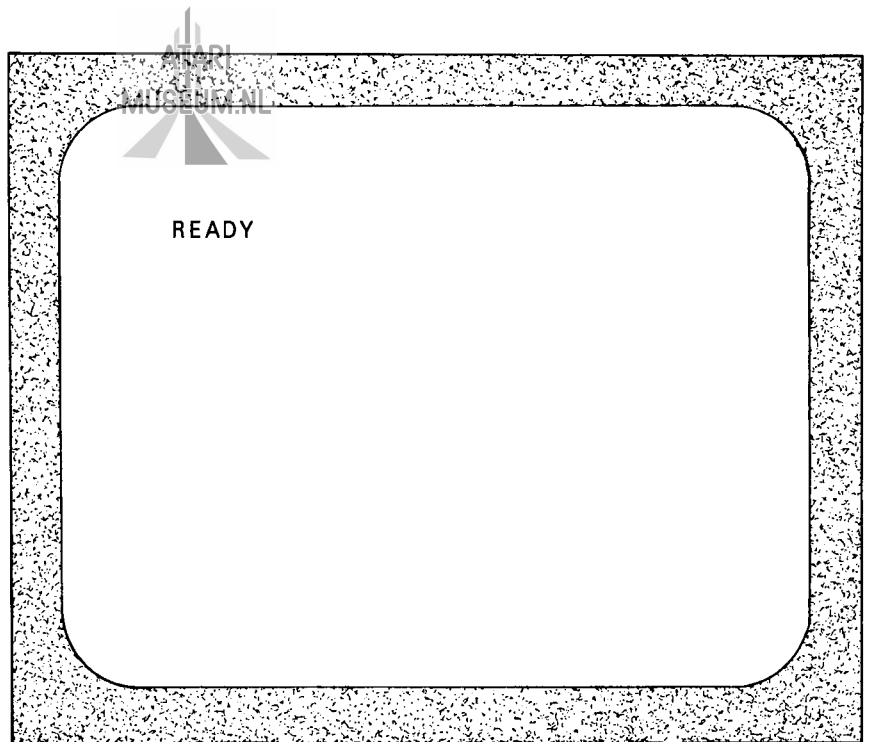
Nadat de computer is afgezet moet 3 tot 5 seconden worden gewacht voordat de computer weer kan worden ingeschakeld.

1.3 Het afstemmen van het tv-kanaal

U ziet nog niets op het tv-scherm, tenzij de televisie reeds was afgestemd op kanaal 4 van de VHF-band (dit is de band waarop normaal ook Nederland 1 is te ontvangen). Is de tv nog niet afgestemd, doe dit dan nu. U moet nu beeld op het scherm krijgen zoals de illustratie op de volgende bladzijde laat zien.

Het woord READY (gereed) geeft aan dat de computer gereed is om aan het werk te gaan.

Het knipperende vierkantje onder de R is de cursor. De cursor geeft aan waar het eerstvolgende teken (letter, cijfer of wat u ook typt) zal verschijnen.



1.4 Het testen van de computer

Het eerste dat nu moet worden gedaan, is het testen van het computersysteem.

Dit wordt gedaan met behulp van het SELF-TEST-menu. (Een **menu** is een bepaalde werkvolgorde die op het beeldscherm wordt getoond.)

Voor het menu wordt aangezet, zullen we eerst enkele toetsen bekijken die nodig zijn voor het testen van de computer. De juiste functie van deze toetsen wordt in hoofdstuk 2 behandeld.

Aan de rechterzijde van de computer bevinden zich van boven naar beneden de volgende toetsen:

RESET-toets (reset = opnieuw aanzetten)

OPTION-toets (option = keuze)

SELECT-toets (select = selecteren)

START-toets

HELP-toets

Tevens bevindt zich op de tweede rij van het toetsenbord (links van de OPTION-toets) de RETURN-toets.

Het aanzetten van het SELF-TEST-programma kan op twee manieren geschieden:

- 1 Indien de computer niet aanstaat: druk de OPTION-toets in, hou deze ingedrukt en schakel de computer in.
- 2 Indien de computer aanstaat: type het woordje BYE en druk op de RETURN-toets.

Het SELF-TEST-menu komt nu op het beeldscherm te staan.

De punten die getest kunnen worden zijn:

MEMORY (het geheugen)

AUDIO - VISUAL (het geluid en tekeningen)

KEY - BOARD (het toetsenbord)

ALL - TEST (alle bovenstaande testen achter elkaar)

Als het menu verschijnt knippert het woord MEMORY. Wilt u een ander gedeelte van het computersysteem testen, dan moet de SELECT-toets

worden ingedrukt tot het te testen deel knippert. Om de test te starten moet de START-toets worden ingedrukt.

Bij de MEMORY-test wordt het geheugen van de computer getest. Het geheugen bestaat uit twee delen, het ROM-geheugen en het RAM-geheugen. Deze geheugendelen worden verderop in het boek besproken.

Zolang een stukje geheugen wordt getest is de testbalk wit. Indien de test uitwijst dat het geheugen correct is wordt de balk geel/groen.

Het ROM-geheugen bestaat uit twee delen, het RAM-geheugen uit 40 delen die alle afzonderlijk worden getest.

Als de test is afgelopen, begint de computer automatisch dezelfde test opnieuw uit te voeren. Wilt u overschakelen naar de volgende test dan moet de HELP-toets worden ingedrukt. Het menu verschijnt opnieuw en door het indrukken van de SELECT-toets kan een ander testgedeelte worden ingeschakeld.

Voor het testen van het AUDIO-VISUAL-gedeelte moet eerst het geluid van de tv worden aangezet. Op het scherm verschijnt nu een notenbalk waarin de geteste noten verschijnen. Alle VOICES (stemmen) van de computer worden achtereenvolgens getest.

Het overschakelen naar het volgende te testen deel geschiedt op de gebruikelijke wijze: eerst de HELP-toets indrukken. Door middel van de SELECT-toets het volgende menu selecteren (KEY-BOARD). De START-toets indrukken.

Op het scherm wordt het toetsenbord getoond. Door een toets in te drukken hoort u een toon en licht de ingedrukte toets op. De toetsen RESET en HELP kunnen niet worden getest daar deze normaal blijven werken.

Door de RESET-toets wordt het normale beeld weer ingeschakeld.

Door de HELP-toets komt het SELF-TEST-menu weer op het scherm te staan.

De toetsen CONTROL en SHIFT werken alleen indien deze toetsen gelijktijdig met een andere toets worden ingedrukt.

De BREAK-toets reageert in het geheel niet.

Nu terug naar het menu door de HELP-toets in te drukken.

De vierde testmogelijkheid is ALL-TEST. Indien deze wordt gekozen (door middel van de SELECT-toets), worden alle voorgaande testen automatisch achter elkaar uitgevoerd.

Door vervolgens op de RESET-toets te drukken krijgt u het normale beeld weer terug.

Nu kan het eerste programma vanaf de cassetterecorder in de computer worden geladen.

1.5 De cassetterecorder

Zoals we in de volgende hoofdstukken zullen zien, zijn programma's instructies die de computer vertellen wat hij moet doen. De opzet van dit boek is te leren hoe u een eigen programma kunt opstellen en dit, met behulp van het toetsenbord, in de computer kunt invoeren.

Als aanvulling op uw eigen programma's kunnen ook door anderen opgestelde programma's of programma's op cassettebandjes worden gebruikt. Bij dit boek zijn twee cassettes met verschillende demonstratieprogramma's gevoegd. Haal de cassette met het programma 'Namen' uit de doos en plaats de cassette in de recorder.

Indien nodig moet het cassettebandje terug worden gespoeld door de REWIND-knop op de recorder in te drukken. Stel vervolgens de bandteller op de recorder op '000' door het knopje naast de teller in te drukken. Het is belangrijk na het afspelen van ieder programma de stand van de teller op te nemen en te noteren op de lijst in Aanhangsel A. Dit vergemakkelijkt het terugzoeken van programma's.

1.6 Het laden van een programma

Het laden van een programma vanaf een cassettebandje in het geheugen van de computer is zeer eenvoudig. Afhankelijk van de lengte van het programma neemt het laden meer of minder tijd. Zet het geluid van de tv aan en type op het toetsenbord:

CLOAD

en druk op de RETURN-toets.

Een pieptoon waarschuwt u dat de PLAY-knop van de cassette recorder moet worden ingedrukt.

Na het indrukken van de PLAY-knop moet weer de RETURN-toets worden ingedrukt. (De pieptoon blijft u horen gedurende het laden.)

Het cassettebandje gaat nu draaien en het programma wordt in het geheugen van de computer geladen.

Wanneer de band stopt omdat het programma is geladen, verschijnt op het scherm:

READY

Het kan zijn dat tijdens het laden van het programma op het scherm het woordje

ERROR

verschijnt.

ERROR betekent fout. Achter het woord ERROR komt meestal een getal te staan dat de soort fout aangeeft. Alle foutmeldingen zijn opgenomen in Aanhangsel B.

In dit geval kunnen het de getallen 138 of 143 zijn die aangeven dat de recorder niet op de juiste wijze werkt. Raadpleeg hiervoor het boekje dat bij de cassette recorder is geleverd.

Taak

1a

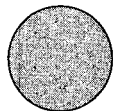
Type op het toetsenbord RUN en druk op de RETURN-toets.

Voer de opdrachten in het programma uit.

Waarschuwing: het uitschakelen van de computer wist automatisch het programma uit het geheugen van de computer. Om het programma weer in het geheugen te brengen moet de LOAD-procedure opnieuw worden uitgevoerd.

Wilt u na het beëindigen van het programma het schone scherm weer terug, dan moet de RESET-toets (rechts bovenaan) worden ingedrukt. Het programma blijft dan gewoon in het geheugen van de computer staan. (Probeer dit door opnieuw RUN te typen en de RETURN toets in te drukken.)

Taak 1a afgerond

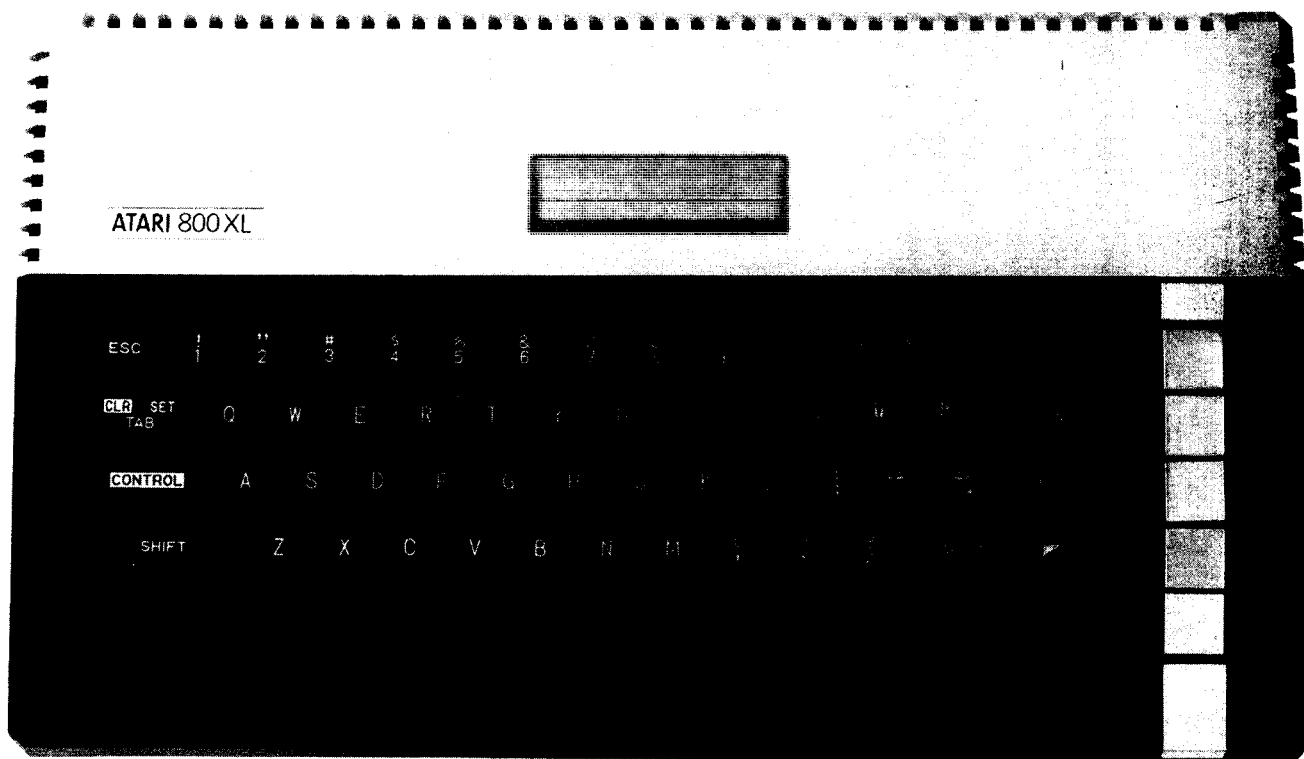


Het toetsenbord

In dit hoofdstuk wordt het toetsenbord behandeld. Door middel van het toetsenbord worden aan de computer opdrachten gegeven. Het is dus noodzakelijk het toetsenbord zo goed mogelijk te leren kennen.

In grote lijnen wijkt het toetsenbord niet veel af van dat van een schrijfmachine. Behalve de meeste toetsen van de schrijfmachine vindt u op de ATARI 800XL ook nog een aantal toetsen die ieder een eigen functie hebben.

Het toetsenbord kan in twee delen worden verdeeld: links het normale toetsenbord en rechts de speciale toetsen die we in het vorige hoofdstuk al hebben gebruikt, zoals de OPTION-, SELECT- en START-toetsen.



2.1 Het linker gedeelte van het toetsenbord

Als u de computer en de tv aanzet, ziet u onder het woord READY een knipperend vierkantje, de cursor. De cursor geeft de plaats aan waar het eerstvolgende teken komt te staan en loopt tijdens het typen mee naar de volgende plaats op het scherm.

Verderop in dit hoofdstuk zien we hoe de positie van de cursor kan worden bestuurd door vier toetsen, zodat de tekst of tekens op iedere willekeurige plaats op het scherm kunnen worden getypt.

De meeste toetsen op het toetsenbord zijn repeterende toetsen. Dit wil zeggen dat als een toets wordt ingedrukt en ingedrukt wordt gehouden, de letter, het cijfer of een ander teken een aantal malen achter elkaar op het scherm wordt afgedrukt.

De ATARI typt erg licht, het is dus voldoende de toets even in te drukken. Probeer nu iets te typen met de letter- en cijfertoetsen. Zoals u ziet krijgt u alleen hoofdletters (de kleine letters worden later in dit hoofdstuk behandeld).

Het kan zijn dat tijdens het typen een ERROR (foutmelding) op het scherm komt te staan, doordat een verkeerde toets werd ingedrukt. Voor hetgeen u nu aan het doen bent is dit onbelangrijk, u kunt rustig doortypen. Het is onmogelijk dat de computer defect raakt door het verkeerde gebruik van de toetsen, het enige dat kan gebeuren is dat de getypte tekst onbegrijpelijk wordt.

Wanneer u aan het einde van een regel bent gekomen, verdwijnt de cursor naar het begin van een nieuwe regel, de getypte tekst lijkt om de rand van het scherm te slaan naar een nieuw beginpunt.

Wat denkt u dat er gebeurt als het scherm wordt vol getypt? Probeer dit eens. Zie hoe de tekst aan de bovenzijde verdwijnt om aan de onderzijde ruimte te maken voor een nieuwe regel. De tekst rolt als het ware omhoog en de bovenste regel is voorgoed verdwenen.

Indien u het scherm schoon wilt maken, is het voldoende de RESET-toets, rechts boven, in te drukken.

Hou nu om beurten enige toetsen ingedrukt om te zien hoe het repeteren van de toetsen werkt.

Onderaan het toetsenbord is de spatiebalk geplaatst. Ook de spatiebalk is repeterend. Door spaties over niet gewenste tekens te typen worden deze tekens gewist.

De cijfer- en leestekentoetsen zijn van twee symbolen voorzien. Het cijfer '1' bijvoorbeeld heeft een ! boven de 1 staan. Als zo'n toets wordt ingedrukt zal het onderste symbool op het scherm komen.

Om het bovenste symbool af te beelden moet een van de SHIFT-toetsen gelijktijdig met de betreffende toets worden ingedrukt. Probeer dit eens.

2.2 Speciale toetsen

Op het toetsenbord zijn een aantal toetsen met speciale functies geplaatst. Deze toetsen worden stuk voor stuk besproken. Tevens zal in het verdere verloop van dit boek nog worden aangegeven wanneer en waarvoor deze toetsen worden gebruikt.

RETURN

Deze toets is op de tweede rij aan de rechterzijde geplaatst. De RETURN-toets is een van de meest gebruikte toetsen. In ieder programma moet deze toets meerdere malen worden ingedrukt om de computer te vertellen dat het getypte stukje in het geheugen van de computer moet worden opgeslagen.

Een heel stel opdrachten aan de computer wordt afgesloten door de RETURN-toets in te drukken (zoals bij het laden van een programma in hoofdstuk 1).

Indien u tijdens het typen van stukjes aan het begin van dit programma de RETURN heeft ingedrukt kwam op het scherm de foutmelding:

ERROR - 17 (Dit betekent: verkeerde gegevens ingevoerd.)

BREAK

Deze toets is rechts boven geplaatst en wordt gebruikt om een lopend programma te onderbreken. In het verdere verloop van dit boek wordt het precieze gebruik van deze toets nader toegelicht.

De CONTROL-toets, links onder op de derde rij, wordt uitsluitend samen met andere toetsen gebruikt. Zet het geluid van de tv aan en druk gelijktijdig op de CONTROL-toets en toets '2'; u hoort nu een zoemtoon.

DELETE/BACK/SPACE

Deze toets is links naast de BREAK-toets geplaatst. Door het indrukken van deze toets wordt de cursor een plaats naar links opgeschoven en het op die plaats staande teken gewist (delete betekent wissen).

Door de CONTROL-toets in te drukken, ingedrukt te houden en daarna op de DELETE-toets te drukken, wordt het teken waarop de cursor staat gewist.

Door een van de SHIFT-toetsen in te drukken en ingedrukt te houden, daarna op de DELETE-toets te drukken, wordt de hele regel waarin de cursor staat gewist.

INSERT

De INSERT-toets (insert is invoegen), links naast de DELETE-toets, wordt ook weer samen met een andere toets gebruikt. Als de CONTROL- en INSERT-toets gelijktijdig worden ingedrukt, ontstaat ruimte op de plaats waar de cursor staat. De tekst naast de cursor schuift naar rechts op.

Door het gelijktijdig indrukken van een SHIFT-toets en de INSERT-toets wordt de hele regel, waarvoor de cursor staat, naar beneden gebracht en kan een tekstregel worden ingevoegd.

CLEAR

Deze toets is naast de INSERT-toets geplaatst. Het gelijktijdig indrukken van de CONTROL-toets of een van de SHIFT-toetsen en de CLEAR-toets maakt het scherm schoon (clear is schoonmaken).

ESC

Deze toets, links bovenaan het toetsenbord, wordt meestal samen met een andere toets ingedrukt. In voorkomende gevallen wordt dit in het boek aangegeven.

De ESC-toets kan ook in programma's worden gebruikt om van het ene programmadeel over te springen naar een ander deel (escape is ontsnappen).

CLR/SET/TAB

De CLR/SET/TAB-toets wordt gebruikt voor het wissen (CLEAR) en het instellen (SET) van de TABulator-instelling.

Indien de toets alleen wordt ingedrukt, zal de cursor verspringen naar de eerstvolgende tabulator-instelling.

Het gelijktijdig indrukken van de SHIFT- en de TAB-toets wist de tabulatorstop.

Indien de CONTROL-toets en de TAB-toets gelijktijdig worden ingedrukt, zal op de plaats waar de cursor staat een tabulatorstop worden aangebracht.

CAPS

De CAPS-toets (capitals zijn hoofdletters), rechts op de tweede rij van onder, zet wanneer de toets wordt ingedrukt, de computer in de kleine-letter-mode.

Het gelijktijdig indrukken van een SHIFT- en de CAPS-toets brengt de computer weer terug in de hoofdletter-mode.

Het indrukken van de CONTROL-toets en de CAPS-toets borgt de CONTROL-toets. De CONTROL-toets blijft als het ware in ingedrukte stand staan.

Om de CONTROL-toets weer in vrije stand te plaatsen, is het voldoende alleen de CAPS-toets in te drukken.

CURSOR

De toetsen waarmee de cursor kan worden verplaatst, zijn de vier toetsen midden rechts die zijn voorzien van de pijlen ↑, ↓, ← en →. De cursor kan door middel van deze toetsen over het hele scherm worden bewogen zonder dat de tekst die op het scherm staat verandert.

Om de cursor omhoog te bewegen moet u de CONTROL-toets ingedrukt houden en daarna op de ↑-toets drukken. Hetzelfde geldt voor het bewegen van de cursor naar beneden: CONTROL + ↓; naar links: CONTROL + ←; naar rechts: CONTROL + →.

Indien de toetsen ingedrukt worden gehouden zal, als de cursor de rand van het scherm heeft bereikt, de cursor om de rand slaan en op de volgende regel aan het andere eind van het scherm weer verschijnen.

De andere tekens op de CURSOR-toetsen kunnen op de normale manier op het scherm worden gebracht.

-, =, + en * door de toetsen gewoon in te drukken.

-, |, \ en ^ door een SHIFT-toets en de betreffende toets gelijktijdig in te drukken.

■ REVERSE VIDEO

Na het indrukken van de REVERSE VIDEO-toets worden alle getypte tekens in de kleur van de achtergrond op het scherm gebracht, omgeven door een blokje in de tekstkleur.

Het opnieuw indrukken van de REVERSE VIDEO-toets schakelt deze mode weer uit.

REVERSE VIDEO wordt in programma's gebruikt om speciale effecten te verkrijgen.

Oefen nu een poosje met het gebruik van de diverse toetsen: hoofdletters, kleine letters, reverse video, wissen en invoegen van tekst en de cursor-bewegingen. Probeer de mogelijkheden goed in u op te nemen. In het begin worden de speciale toetsen nog wel aangegeven, maar het werkt prettiger indien u ze goed kunt vinden.

De onderstaande toetsen, die aan de rechterzijde zitten, zijn we al tegengekomen in het eerste hoofdstuk. Deze toetsen worden verderop in het boek uitgebreid behandeld.

RESET

Het indrukken van de RESET-toets maakt het scherm schoon en brengt het beginbeeld op het scherm.

OPTION

Door de OPTION-toets te gebruiken, kan een keuze worden gemaakt uit verschillende programmavariaties.

SELECT

De SELECT-toets dient om in een programma een selectie te maken uit diverse toepassingen.

De **START**-toets wordt gewoonlijk gebruikt om een programma of een spel op gang te brengen.

HELP

De **HELP**-toets kan in een programma worden opgenomen om hulp te bieden bij sommige keuzemogelijkheden.

Taak

2a

Laad nu het tweede programma van de oefencassette, 'Letters tikken'. Indien u het bandje na het eerste programma niet heeft teruggespoeld, kan direct met het laden worden begonnen zoals in hoofdstuk 1 is beschreven. Indien het bandje wel is teruggespoeld, moet de teller op 000 worden gezet en daarna de **ADVANCE**-knop op de cassetterecorder worden ingedrukt tot de teller de door u genoteerde stand heeft bereikt. Laad hierna het programma zoals in hoofdstuk 1 is beschreven. Noteer na het stoppen van de cassetteband wederom de stand van de teller in Aanhangsel A.

Voer de opdrachten uit die in het programma worden gegeven.

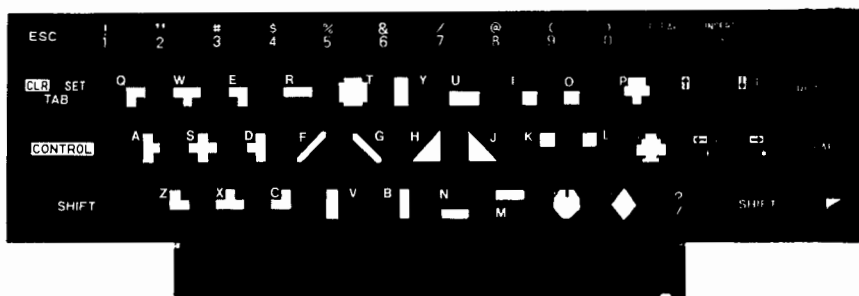
Taak 2a afgerond

2.3

De grafische symbolen

Op het toetsenbord zijn negenentwintig grafische symbolen opgenomen. Met behulp van deze tekens kunnen tekeningen en grafieken worden gemaakt of tabellen worden opgesteld.

De symbolen zijn op onderstaande wijze in de toetsen opgenomen.



De symbolen kunnen op het scherm worden gebracht door de **CONTROL**-toets ingedrukt te houden en dan op de betreffende toets te drukken.

Indien u veel symbolen achter elkaar op het scherm wilt zetten, kan de **CONTROL**-toets vast worden gezet door zowel de **CONTROL**- als **CAPS**-toets gelijktijdig in te drukken.

De grafische symbolen kunnen ook in reverse video worden afgedrukt. Dit is bijvoorbeeld belangrijk als een teken wordt gezocht dat schijnbaar niet in de symbolen is opgenomen. ▀ bijvoorbeeld is ▾ in reverse video. Probeer de verschillende tekens eens te typen en type daaronder de reverse video van deze tekens.

Instructies

In dit hoofdstuk leren we de computer opdrachten te geven voor het automatisch afdrukken van teksten op het scherm en het uitvoeren van eenvoudige berekeningen.

De opdrachten worden aangeduid als:

- a **Instructies**, indien ze in een programmaregel worden opgenomen (dit heet de 'programmeer-mode');
- b **Commando's**, indien ze direct op het scherm worden getypt (dit is de 'directe mode').

Een en ander zal u in de loop van dit hoofdstuk wel duidelijk worden.

ATARI BASIC bestaat in hoofdzaak uit eenvoudige instructies en commando's, meestal slechts één woord, die de computer vertellen wat er moet worden gedaan.

Instructies en commando's worden door de computer alleen geaccepteerd indien ze in **hoofdletters** zijn geschreven. Voorlopig zullen we dus de computer, voor alles wat getypt wordt, in de hoofdletter-mode laten staan.

Na het typen van een opdracht op het scherm moet altijd de RETURN-toets worden ingedrukt om óf de instructie in het geheugen van de computer op te nemen, óf de computer het commando uit te laten voeren.

Er zijn enige instructies die ook in de directe mode kunnen worden gebruikt, zoals de PRINT-instructie. Deze worden in dit boek echter met de naam instructie aangeduid.

3.1 De PRINT-instructie

Print betekent afdrukken. De PRINT-instructie is een van de meest krachtige instructies. PRINT geeft aan dat alles wat rechts van deze instructie staat op het scherm moet worden afgedrukt of, bij berekeningen, moet worden uitgevoerd.

De PRINT-instructie is erg flexibel en kan verschillende vormen hebben. Probeer elk van de volgende instructies te typen; vergeet niet aan het einde van iedere instructie de RETURN-toets in te drukken om de computer te vertellen dat u een antwoord verwacht.

Voorlopig zal als geheugensteuntje, totdat het een automatische handeling is geworden, achter iedere opdracht (RETURN) staan.

Type nu:

```
PRINT "DE EERSTE POGING" (RETURN)
```

Probeer nu

```
PRINT 180+35 (RETURN)
```

U ziet dat na de eerste PRINT-instructie het gedeelte tussen de dubbele aanhalingstekens op het scherm verscheen, terwijl na de tweede PRINT-instructie het resultaat van de optelling werd getoond.

PRINT "180+35"

Deze maal verschijnt op het scherm 180+35 in plaats van 215, het resultaat van de optelling 180+35.

Over het algemeen zal op het scherm alles dat tussen dubbele aanhalingstekens is geplaatst, worden afgedrukt zoals de schriftekens daar staan (in dit geval dus "180+35"). Als de aanhalingstekens worden weggelaten zal de computer eerst de bewerking uitvoeren en daarna het resultaat tonen.

Voordat we nu wat gaan oefenen moeten we het eerst over de decimale punt hebben. De komma in getallen zoals 2,25 moet bij gebruik voor de computer worden vervangen door een punt: 2,25 wordt dus 2.25. Deze schrijfwijze komt voort uit de Engelse afkomst van de computer. Een punt in een gebroken getal wordt de 'decimal point' genoemd. Ook de nul voor de punt wordt dikwijls weggelaten; 0.45 kan dus ook als .45 worden getypt.

Uit het bovenstaande volgt dat bij grote getallen onze schrijfwijze 5.345.832 niet mag worden gebruikt, dit moet gewoon 5345832 worden. De computer begrijpt de eerste schrijfwijze niet en maakt er een gebroken getal van.

Een tweede afwijking bij computerbewerkingen is het vermenigvuldigings-teken. Daar wordt het teken * voor gebruikt (op de CURSOR-toets). Het voor ons normale teken '×' zal door de computer als de letter X worden gelezen.

Dit geldt ook voor het deelteken :. Dit wordt door de computer alleen begrepen als een dubbele punt. Als deelteken gebruikt de computer / (onder het vraagteken naast de rechter SHIFT-toets).

Het kan zijn dat u in de PRINT-instructie een fout typt die de computer niet begrijpt, bijvoorbeeld de aanhalingstekens na de PRINT-instructie vergeet, of het woord PRINT verkeerd typt. U krijgt dan de foutmelding: ERROR - met daarachter de regel waarin de fout zit en de vermoedelijke plaats van de fout in reverse video. Makkelijker kan het al niet. En nu aan het werk met de volgende taak.

Taak

3a

Type de volgende PRINT-instructies op de computer. Probeer eerst het resultaat te raden alvorens de RETURN-toets in te drukken.

PRINT 10 * 25	(RETURN)
PRINT "10+25"	(RETURN)
PRINT 10+0.25+5	(RETURN)
PRINT 225 / 25	(RETURN)
PRINT "225 / 25"	(RETURN)
PRINT "HALLO"	(RETURN)
PRINT HALLO	(RETURN)

Taak 3a afgerond





U bent misschien verrast door het resultaat van de laatste PRINT-instructie. Daar voor en achter het woord geen aanhalingstekens waren geplaatst, wordt de instructie door de computer als een berekening gelezen en werd het resultaat van deze berekening afgedrukt.

Zoals u ziet is de nul voorzien van een streep /. Dit is om te voorkomen dat de letter O verwisseld wordt met het cijfer 0.

3.2 Computergeheugen

Nu eerst iets over het geheugen van de computer, en wel dat deel dat wordt gebruikt om eigen informatie in op te slaan.

In hoofdstuk 1 heeft u bij het testen van de computer al gezien dat er twee geheugendelen zijn. Het deel waarin eigen informatie wordt opgeslagen, is het RAM (Random Access Memory), dat willekeurig toegankelijk geheugen betekent.

Dit geheugen kunnen we voorstellen als een hoeveelheid lege dozen waarin we naar willekeur informatie kunnen opbergen. Deze informatie blijft in het geheugen tot de computer wordt afgezet of op een andere wijze het geheugen wordt gewist. Een van de manieren waarop we het geheugen leeg kunnen maken, wordt verderop in dit hoofdstuk behandeld.

Het geheugen van de ATARI 800XL is vrij groot, namelijk 64K. 64K wil zeggen dat het geheugen 64 kilobytes ruimte heeft. Een byte is de geheugenruimte die we nodig hebben om een teken (letter, cijfer, grafisch symbool) in op te slaan.

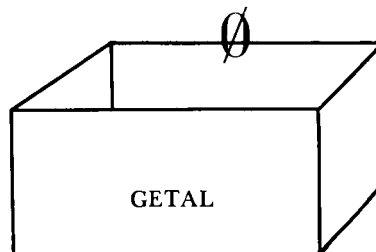
64 kilobytes is dus 64000 bytes (eigenlijk nog meer, want 1 kilobyte is 1024 bytes). Het aantal tekens op een geheel vol getypt scherm is 960.

3.3 Variabelen

Nu weer even terug naar de opdracht PRINT HALLO.

Anders dan in de eerste instructies bevatte de reeks schriftekens waarvan de waarde moest worden bepaald geen getallen en bewerkingstekens zoals + of -. In zo'n geval behandelt de computer de tekst als een **variabele**.

Een variabele kan worden beschouwd als informatie die we in een lege doos van het computergeheugen stoppen.

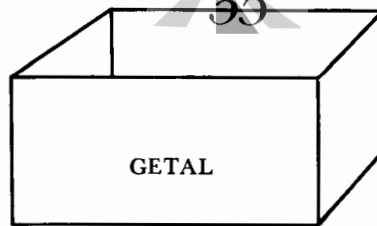


Omdat in de variabele HALLO geen getal voorkwam, drukte de computer als resultaat 0 af. Om een getal in een variabele te plaatsen (we noemen dit de waarde aan een variabele toekennen), gebruiken we een eenvoudige instructie in onderstaande vorm.

```
LET HALLO=35      (RETURN)
```

Probeer dit en type daaronder:

```
PRINT HALLO      (RETURN)
```



Deze maal moet het antwoord van de computer 35 in plaats van 0 zijn zoals het geval was in taak 3a. We hebben nu als waarde voor de variabele 35 in het geheugen geplaatst.

LET... = ... betekent LAAT... ZIJN...

Ook het woordje LET is een BASIC-instructie en wordt dus alleen als LET door de computer begrepen.

Een variabele heeft altijd een naam. In dit geval was de naam HALLO, maar het had ook A, X, AB of A1 kunnen zijn, zolang als het eerste teken maar een letter is. Dus **nooit** 1A, 2AB, 3HUIS enzovoort.

3.1 Soorten variabelen

Aan een variabele kan als waarde een getal of een bewerking met getallen worden toegekend zoals in:

```
LET A=15
LET PRIJS=3*18-2
```

Deze variabelen worden **numerieke variabelen** genoemd.

Aan een variabele kan ook een waarde in letters of symbolen worden toegekend. Om deze variabele te onderscheiden van de numerieke variabele wordt hij de **string-variabele** genoemd.

Een string is in BASIC een reeks schriftekens (woorden, symbolen, letters, cijfers of combinaties daarvan). De string-variabele wordt altijd tussen aanhalingstekens geplaatst.

Aan de naam van de string-variabele (het gedeelte voor het = teken) wordt, ter onderscheiding van de numerieke variabele, het dollarteken \$ toegevoegd (SHIFT + 4-toets).

De namen hebben dus deze vorm: HUIS\$, A\$, A3\$, B\$, BEEST\$ enzovoort (spreek uit huis-string, a-string enzovoort).

De computer leest de naam en plaatst deze in het geheugen. In het begin is het – vooral voor de duidelijkheid in een programma – aan te raden een naam te kiezen die iets van de variabele vertelt, zoals:

- a De prijs van koffie = 4.20. Variabele is dan KOFFIE of PRIJS.
- b De hoofdstad van Nederland = Amsterdam. Variabele is dan HOOFDSTAD\$. U kunt dat ook afkorten tot b.v. HFDST\$.

Het enige waarop moet worden gelet is dat in een programma nooit twee gelijke namen mogen voorkomen voor verschillende variabelen. Dus nooit:

```
LET A=12 en
LET A=24+3
of
LET BEEST$="KOE" en
LET BEEST$="KAT"
```

Er zijn nog enige namen die niet mogen worden gebruikt zoals de namen van instructies en commando's (PRINT, LET enz.). Gebruik daarom het liefst Nederlandse benamingen of lettercombinaties.

Een tweede punt van belang voor de string-variabele is dat de computer moet weten hoeveel ruimte in het geheugen moet worden gereserveerd voor de variabele. Dit heet het DIMensioneren van de string. Het reserveren van de ruimte wordt gedaan door de DIM-instructie en gaat als volgt:

```
DIM HOND$(6)
LET HOND$="POEDEL"
PRINT HOND$
```

In dit geval heeft u zes schriftekens gedimensioneerd. Indien de hond een sint-bernhard zou zijn, komt u ruimte te kort. In dit geval had tussen de haakjes 13 moeten staan.

U hoeft voor iedere DIM-instructie niet het juiste aantal letters of tekens te tellen, voor POEDEL is DIM HOND\$(10) ook prima. Meer mag, maar minder nooit.

Meer wil niet zeggen dat u steeds maar 100 of 1000 tussen haakjes moet zetten, want dit kost allemaal geheugenruimte. Voor een kort programma is dit niet belangrijk, maar voor lange programma's kan dit wel eens tekorten in het geheugen veroorzaken.

3.5 Het scherm schoonmaken

Het scherm schoonmaken kan op twee manieren geschieden:

- 1 Door de RESET-toets in te drukken. Het scherm komt in de beginstand te staan: READY met daaronder de cursor.
- 2 Door de CONTROL-toets en de CLEAR-toets gelijktijdig in te drukken. Het scherm wordt dan leeg op de cursor links boven na.

Maak het scherm schoon.

We gaan nu verder met het geven van een aantal instructies aan de computer. Type de instructies nauwkeurig over – ook de spaties.

Indien (wat waarschijnlijk is, tenzij u een kei bent in het typen) u een ERROR-melding krijgt, kan het beste de regel gewoon over worden getypt. Voor de betekenis van de eventuele nummers achter het woord ERROR wordt verwezen naar Aanhangsel B.

(RETURN) op iedere regel wil zeggen: druk de RETURN-toets in.

Type nu:

```
DIM A$(3) (RETURN)
DIM BOOM$(10) (RETURN)
DIM K$(8) (RETURN)
DIM DIER$(6) (RETURN)
DIM H$(10) (RETURN)

LET A$="EEN" (RETURN)
LET B=12+4 (RETURN)
LET BOOM$="APPELBOOM" (RETURN)
LET C=41-3 (RETURN)
LET K$=" IN EEN " (RETURN)
LET DIER$=" PAARD" (RETURN)
```



```
LET SOM=3000+1 (RETURN)
LET H$="HOTEL" (RETURN)
LET DELING=500/50 (RETURN)
LET X=12*4/6 (RETURN)
```

Let op: vergeet de spaties voor en achter IN EEN en vóór PAARD niet!

Taak

3b

Noteer alvorens de PRINT-instructie te typen datgene wat op het scherm komt te staan.

```
PRINT A$ (RETURN)
PRINT C (RETURN)
PRINT K$ (RETURN)
PRINT SOM (RETURN)
PRINT H$ (RETURN)
PRINT X (RETURN)
```

Taak 3b afgerond



Zet de computer nog niet uit, want er is meer te doen met de variabelen die in het geheugen van de computer zijn opgeslagen.

3.6 Het aaneenschakelen van PRINT-instructies

Tot nu toe hebben we voor iedere PRINT-instructie een nieuwe regel gebruikt. We zullen nu bekijken hoe de PRINT-instructies kunnen worden gekoppeld, zodat de verschillende strings achter elkaar worden getoond.

Maak het scherm schoon (door de RESET-toets of CONTROL + CLEAR). Type de volgende instructie:

```
PRINT A$;DIER$;K$;BOOM$ (RETURN)
```

De puntkomma achter iedere variabele vertelt de computer dat u iedere string direct achter de voorgaande op het scherm wilt zien.

Indien zo'n instructie wordt gebruikt moet van tevoren rekening worden gehouden met de ruimte tussen de strings. In de LET-instructies moeten de spaties worden opgenomen tussen de aanhalingstekens, anders komt een onleesbaar geheel op het scherm te staan, zoals in onderstaand voorbeeld.

Type:

```
PRINT A$;BOOM$ (RETURN)
```

Indien numerieke variabelen worden gekoppeld, komen alle getallen achter elkaar te staan. Probeer dit maar eens door het volgende te typen:

```
PRINT B;C;G1;X (RETURN)
```

Het aaneenkoppelen van getallen is echter nooit nodig, dus dit geeft geen problemen.

Oefen een poosje met LET- en PRINT-instructies en gebruik daarbij verschillende getallen, strings (vergeet niet te DIMensioneren), teksten en getallen tussen aanhalingstekens en de puntkomma voor het aaneenschakelen van strings.

3.7 Het NEW-commando

We hebben gezien dat het scherm schoon kan worden gemaakt door RESET of CLEAR. Het probleem is dat desondanks alle ingebrachte variabelen in het geheugen van de computer aanwezig blijven.

Het schoonmaken van het geheugen kan op twee manieren geschieden:

- 1 Zet de computer uit en schakel hem na 5 seconden weer in.
- 2 Type het woordje NEW en druk op de RETURN-toets.

NEW (nieuw) is een commando dat alle door u ingebrachte gegevens uit het geheugen van de computer verwijdert. Het is definitief, dus wees voorzichtig met het gebruik van dit commando. Pas als u zeker weet de gegevens niet meer nodig te hebben typt u:

NEW (RETURN)

en het geheugen is gewist.

De tekst die op het scherm staat blijft echter gewoon staan en kan worden verwijderd door het scherm schoon te maken (RESET of CLEAR).

Taak

3c

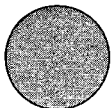
Wis het geheugen en maak het scherm schoon.

Maak drie LET-instructies zodat na het aaneenschakelen in een PRINT-instructie (door de puntkomma) de volgende zin op het scherm verschijnt:

IK KAN AL OVERWEG MET DE COMPUTER

Eerst dus drie DIM-instructies, dan drie LET-instructies (denk aan de spaties op de juiste plaatsen) en vervolgens de PRINT-instructie.

Taak 3c afgerond



3.8 Kleine letters in de PRINT-instructie

Tot nu toe hebben we steeds hoofdletters gebruikt. In het begin van dit hoofdstuk hebben we geleerd dat de computer alleen hoofdletters als opdrachten accepteert. Dat klopt in zoverre dat de instructies en commando's steeds in hoofdletters moeten worden getypt. Tussen de aanhalingstekens mogen echter gerust kleine letters worden gebruikt. Type:

PRINT "ook dit is mogelijk" (RETURN) ,

Na de eerste aanhalingstekens moet u dus de CAPS-toets indrukken. Om daarna weer in hoofdletter-mode te komen moet u een SHIFT- en de CAPS-toets gelijktijdig indrukken.

Het is ook mogelijk de tekst in de PRINT-instructie in reverse video af te beelden. Dit kan later in programma's een leuk effect geven of verhelderend werken.

Indien u in een PRINT-instructie het **█**-teken tegenkomt wil dat zeggen: druk de toets REVERSE VIDEO in.

Type nu:

```
PRINT "█EEN PRACHTIGE TITEL█" (RETURN)
```

3.9 Printen van grafische symbolen

Ook grafische symbolen kunnen in een PRINT-instructie worden opgenomen.

De in te typen letters worden in dit boek met een streepje | ervoor aangegeven. Dus |F|G wil zeggen: hou de CONTROL-toets ingedrukt en druk op de F-toets en daarna op de G-toets.

Type:

```
PRINT " |I|K|I|K|I|K ATARI|L|O|L|O|L|O" (RETURN)
```

We kunnen zelfs hele tekeningen bestaande uit grafische symbolen – zoals het huisje uit hoofdstuk 2 – in string-variabelen onderbrengen, en door de computer op het scherm laten plaatsen door middel van een PRINT-instructie.

Dit is echter iets dat veel gemakkelijker in een programma kan worden ondergebracht. In het hoofdstuk over programma's komen we hier nog op terug.

3.10 De lengte van de PRINT-instructie

De PRINT-instructie is aan een maximale lengte gebonden. Het totaal aantal tekens van de instructie, inclusief het woord PRINT, mag nooit meer dan drie regels op het scherm bedragen. Zodra de instructie langer is zal alles dat buiten deze drie regels valt niet in het geheugen van de computer worden opgenomen en dus niet op het scherm worden afgedrukt.

Type:

```
PRINT"DIT IS HET MAXIMUM DAT IN EEN KEER  
KAN WORDEN GEPRINT ALS DE INSTRUCTIE TE  
LANG WORDT KOMT HET LAATSTE DEEL NIET O  
P HET SCHERM" (RETURN)
```

U ziet dat het stuk "NIET OP HET SCHERM" niet in het geheugen is opgeslagen.

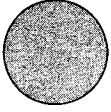
Wat te doen indien u een langer verhaal af wilt drukken? Heel eenvoudig: gebruik in een programma meerdere PRINT-instructies die door de puntkomma aaneen worden geschakeld.

Taak

3d

Laad het derde programma van de oefencassette, 'Vragen'. RUN dit programma en voer de in het programma gegeven opdrachten uit.

Taak 3d afgerond



Noteer wederom de standen van de teller op de lijst in Aanhangsel A. Dit programma is een quizprogramma waarin vragen worden gesteld over de hoofdstukken 2 en 3. Probeer deze vragen te beantwoorden en herhaal de stof uit de hoofdstukken van de vragen waarin u fouten heeft gemaakt.

Programma's en lussen

In dit hoofdstuk leren we hoe instructies tot programma's worden samengevoegd. Tevens bekijken we een van de eenvoudigste programma-instructies, de **programma-lus**.

In hoofdstuk 3 hebben we twee instructies geleerd: de PRINT- en de LET-instructie. We zagen op welke wijze de LET-instructie wordt gebruikt om de waarde aan een variabele toe te kennen.

```
LET PRIJS=235.50           (RETURN)
of
DIM GELD$(7)
LET GELD$=" GULDEN"      (RETURN)
```

Daarna zagen we hoe de waarden van de variabelen op het scherm worden gebracht door:

```
PRINT PRIJS;GELD$        (RETURN)
```

Nu zullen we zien hoe deze instructies in een programma worden ondergebracht.

4.1 Programma's

Een programma is een samenstelling van instructies die in een bepaalde volgorde moeten worden uitgevoerd. Om de volgorde van het programma te bepalen wordt aan iedere instructie een nummer gegeven, het **regelnummer**. Het regelnummer zorgt ervoor dat de instructie op die regel in de juiste volgorde, ten opzichte van de overige instructies in het programma, wordt uitgevoerd.

Als voorbeeld het volgende. Type:

```
10 LET BEDRAG=100           (RETURN)
20 LET BTW=0.19*BEDRAG     (RETURN)
30 PRINT "BTW-BEREKENING"  (RETURN)
40 PRINT                    (RETURN)
50 PRINT "ALS HET BEDRAG= ";BEDRAG (RETURN)
60 PRINT "DAN IS DE BTW ";BTW (RETURN)
70 END                      (RETURN)
```

Dit is een programma dat de BTW (19%) berekent over 100 gulden. Voor zo'n bedrag natuurlijk een onnodig programma, maar later in dit hoofdstuk wordt het programma uitgebreid en wordt het zinniger.

We zullen nu het programma regel voor regel bekijken.

10 geeft de waarde aan de variabele BEDRAG.

20 geeft aan de BTW de waarde van 19%.

30 zet de titel boven het programma.

40 is een lege instructie waardoor het programma een regel overslaat (te gebruiken om een programma op het scherm duidelijker te maken).

50 plaatst de waarde van de variabele BEDRAG achter de uitleg ALS HET BEDRAG =.

60 idem voor de waarde van de variabele BTW.

70 de END-instructie, die het programma laat stoppen.

In dit programma is de END-instructie eigenlijk niet nodig, want een programma stopt normaal na de laatste regel. In sommige programma's moet echter een END-instructie worden opgenomen om het programma te stoppen. Voor alle duidelijkheid nemen we voorlopig in alle programma's de END-instructie op.

De computer heeft nu een eenvoudig programma in het geheugen opgeslagen. Om dit programma op het scherm te brengen bestaat een commando RUN. (Dit bent u al tegengekomen bij de programma's op het oefencassettebandje).

Type:

```
RUN                                (RETURN)
```

en kijk wat er gebeurt.
RUN is het uitvoeringscommando.

De instructies die in programma's zijn opgenomen, worden altijd uitgevoerd in numerieke volgorde, het laagste nummer het eerst. De reden dat de regels in tientallen worden genummerd is eenvoudig: indien nodig kunnen we later een regel bijvoegen die dan in het programma wordt opgenomen.

We kunnen bijvoorbeeld het programma eerst het scherm schoon laten maken. Het is prettiger als het resultaat van het programma op een schoon scherm komt te staan.

Om in een PRINT-instructie de opdracht te geven het scherm te wissen, moet eerst het woord PRINT worden getypt, dan de aanhalingstekens, daarna eenmaal op de ESC-toets drukken, vervolgens de CONTROL- en de CLEAR-toets gelijktijdig indrukken en daarna de aanhalingstekens sluiten.

```
PRINT "\
```

Het pijltje geeft aan dat de ESC-toets is ingedrukt en daarna de CONTROL- en CLEAR-toetsen.

Onthou deze regel goed, want het is overzichtelijk om een programma op een schoon scherm te krijgen. We zullen deze methode dan ook veel toepassen.

Type:

```
5 PRINT "\                                (RETURN)
```

De regel 5 is nu opgenomen in het programma. Hoewel deze programmaregel na de andere regels is getypt, plaatst de computer de regel in de juiste numerieke volgorde. Daar dit regelnummer het laagste is zal deze regel het eerst worden uitgevoerd.

Type nu:

```
RUN                                (RETURN)
```

en het resultaat staat op een schoon scherm.

U kunt de juiste wijze van het invoegen van regel 5 controleren door een ander BASIC-commando te gebruiken: het LIST (lijst)-commando.
Type:

```
LIST (RETURN)
```

Het resultaat van dit commando ziet u nu op het scherm: het hele programma in de juiste volgorde.

Het LIST-commando kan ook worden gegeven om aan te geven welk gedeelte of welke regel(s) van het programma we willen zien. Probeer de volgende LIST-commando's.

```
LIST 5 (RETURN)
LIST 10 (RETURN)
LIST 30, 50 (RETURN)
LIST 50, 150 (RETURN)
```

LIST 5 toont regel 5.

LIST 10 toont regel 10.

LIST 30,50 toont alle regels vanaf 30 tot en met 50.

LIST 50,150 toont in dit geval de regels 50 tot en met 70, daar 70 de laatste regel is. Deze methode wordt gebruikt indien u niet zeker bent van de lengte van een programma en alleen geïnteresseerd bent in het gedeelte dat na regel 50 komt.

4.3 De GOTO-instructie

De BASIC-instructie die we nu gaan leren is de GOTO-instructie. GOTO wil zeggen GA NAAR. Met deze instructie kunnen we in een programma hetzelfde werk steeds opnieuw laten uitvoeren.

Voeg de volgende regel aan het programma toe:

```
65 GOTO 20 (RETURN)
```

Geef het commando:

```
RUN (RETURN)
```

en kijk wat er gebeurt.

U kunt het programma laten stoppen door de CONTROL-toets en de 1-toets gelijktijdig in te drukken.

Om het programma weer te laten lopen moeten wederom beide toetsen gelijk worden ingedrukt.

Als het programma u gaat vervelen kunt u het onderbreken door de BREAK-toets in te drukken (rechts bovenaan).

Op het scherm komt nu te staan:

```
STOPPED AT LINE . . . (gestopt in regel...)
```

De GOTO-instructie werkt als volgt:

Zodra de computer de regels 5 tot en met 65 heeft afgewerkt zal de GOTO-instructie de computer opnieuw laten beginnen bij regel 20, zoals in het volgende schema is te zien.

```

5 PRINT " "
10 LET BEDRAG=110
20 LET BTW=0.19*BEDRAG
30 PRINT "BTW-BEREKENING"
40 PRINT
50 PRINT "ALS BEDRAG= " ; BEDRAG
60 PRINT "DAN IS DE BTW " ; BTW
65 GOTO 20
70 END

```

dit gedeelte wordt
doorlopen tot het
programma wordt gestopt

Het gebruik van de GOTO-instructie veroorzaakt een zogenaamde **lus**. Deze lus laat het programma doorlopen totdat het programma van buitenaf wordt gestopt. Stop het lopende programma door de BREAK-toets in te drukken.

4.4 Het CONT-commando

Wilt u na het indrukken van de BREAK-toets een programma weer verder laten lopen, dan is daarvoor een speciaal commando: het CONT-commando. (Cont = continue dat voortzetten betekent.) Type het woord CONT en druk op de RETURN-toets: het programma gaat nu verder op het punt waar het door de BREAK-toets was onderbroken.

We zullen nu het programma zodanig wijzigen dat na iedere keer dat de lus doorlopen is iets anders op het scherm komt te staan.
Type:

```
62 LET BEDRAG=BEDRAG+5 (RETURN)
```

LIST nu het programma en u zult zien dat de nieuwe regel op de juiste plaats staat. (RETURN na LIST niet vergeten?)

Regel 62 is een speciaal geval van de LET-instructie. Wanneer de computer deze regel uitvoert wordt steeds de waarde 5 opgeteld bij de variabele BEDRAG. Deze wordt dus bij iedere lus groter. Programma's kunnen natuurlijk ook instructies bevatten die het programma laten beëindigen zonder dat er van buitenaf wordt ingegrepen. Deze instructies worden in een volgend hoofdstuk behandeld.

Type nu RUN, druk op de RETURN-toets en laat het programma even lopen. Stop het programma door de BREAK-toets in te drukken. Laat nu het programma weer lopen door het commando CONT te gebruiken. Stop het programma opnieuw.

Maak, alvorens aan taak 4a te beginnen het geheugen schoon door het NEW-commando te gebruiken (NEW typen en RETURN-toets indrukken).

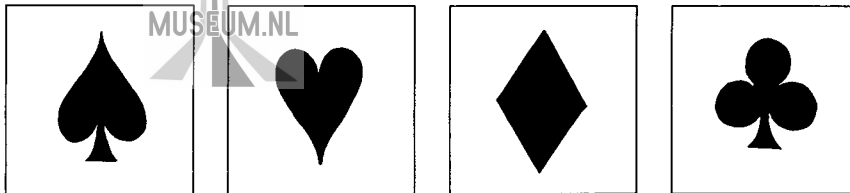
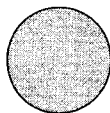
Taak

4a

Bekijk het volgende programma en:

- a noteer in welke volgorde het programma verloopt;
- b geef aan wat er op het scherm komt te staan;
- c type en RUN het programma.

Taak 4b afgerond



4.6 De cursorbewegingen

Niet alleen grafische symbolen kunnen in een PRINT-instructie worden opgenomen, ook de cursorbewegingen kunnen binnen de aanhalingstekens van een PRINT-instructie worden geplaatst. Met die methode is het mogelijk de tekst of een tekening naar rechts, naar links, naar boven of naar beneden te verplaatsen.

Het plaatsen van de cursorbeweging in de PRINT-instructie is gewoon een kwestie van toetsen indrukken in een bepaalde volgorde.

Na de aanhalingstekens: druk eenmaal op de ESC-toets, druk daarna gelijktijdig de CONTROL- en de betreffende cursortoets in. Na de aanhalingstekens verschijnt dan een pijltje op het scherm dat de ingedrukte cursortoets aangeeft: ↑, ↓, ← of →.

In dit boek wordt dit ook op dezelfde wijze in de programma's aangegeven. ↓ wil dus zeggen: ESC-toets indrukken, CONTROL-toets en cursor ↓-toets gelijktijdig indrukken.

Voor iedere cursorbeweging moet deze handeling worden herhaald. Indien dus twee bewegingen achter elkaar in één PRINT-opdracht staan, moeten steeds alle drie de toetsen worden ingedrukt.

Type nu:

```

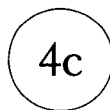
1 0 PRINT "\ " (RETURN)
2 0 PRINT "CURSORBEWEGINGEN" (RETURN)
3 0 PRINT "↓DEZE BEWEGING SLAAT EEN REGEL
OVER" (RETURN)
4 0 PRINT "↓↓→→NAAR BENEDEN EN NAAR RECHT
S" (RETURN)
5 0 END (RETURN)

```

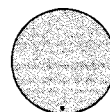
Type RUN (RETURN)

U ziet dat deze methode erg geschikt is indien u in een programma een regel over wilt slaan zonder de lege PRINT-instructie te gebruiken (zoals in de eerste programma's in dit hoofdstuk).

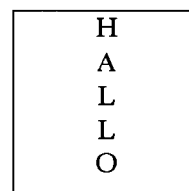
Taak



Taak 4c afgerond



Maak een programma waarin:
 In regel 10 het scherm wordt schoongemaakt;
 In regel 20 het woord HALLO verticaal op het scherm wordt getoond (door gebruik te maken van de cursorbewegingen in deze PRINT-instructie).



ATARI
MUSEUM.NL
RETURN - RETURN - RETURN - RETURN

Na dit hoofdstuk zal het geheugensteuntje (RETURN) na iedere regel worden weggelaten. Aangenomen wordt dat u het indrukken van de RETURN-toets na iedere regel of ieder commando nu automatisch doet. Type en RUN om het niet te vergeten het onderstaande programma.

```
10 PRINT " DENK AAN DE R E T U R N TOETS "  
20 GOTO 10
```

Stoppen door middel van de BREAK-toets.

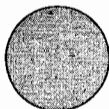
Taak

4d

Laad het programma 'Het schip' van de oefencassette.
RUN dit programma en probeer het schip in de juiste volgorde samen te stellen.

Druk eerst de toetsen 1, 2, 3, 4 en 5 in deze volgorde in. Bekijk de onderdelen en probeer door een andere toetsenvolgorde de juiste tekening van het schip op het scherm te krijgen.

Taak 4d afgerond



Het verbeteren van fouten en het SAVEN (wegschrijven) van programma's

In dit hoofdstuk wordt het verbeteren van fouten in programma's en het opnemen van programma's op cassettebandjes behandeld.

5.1 Het verbeteren van fouten

Vergissen is menselijk en ook u zult herhaaldelijk fouten maken bij het intypen van programma's; en van deze fouten leren. De computer zal u in de meeste gevallen op de fout attent maken door een ERROR-melding te geven.

Een overzicht van deze foutmeldingen vindt u in Aanhangsel B.

Het herstellen van fouten is bij de ATARI 800XL gelukkig een eenvoudige zaak.

Door gebruik te maken van de cursortoetsen, de INSERT-toets en de DELETE/BACK/SPACE-toets kunnen op iedere plaats in de tekst wijzigingen worden aangebracht.

Door het intypen van een regelnummer kan een bestaande regel in zijn geheel worden vervangen.

We zullen het verbeteren van fouten eens nader bekijken aan de hand van enige voorbeelden.

Type:

```
10 PRINT "OPSLAAN IN DE DERDE KOLOM"
```

Indien we nu DERDE willen veranderen in VIJFDE moeten de volgende handelingen worden uitgevoerd:

- a Breng de cursor op de D van DERDE.
- b Type het woord VIJFDE.
- c Druk eenmaal op de CONTROL- en INSERT-toets (dit om ruimte te maken voor de extra letter).
- d Druk op de RETURN-toets.

Belangrijk: Na iedere verandering in een regel van een programma moet ook weer de RETURN-toets worden ingedrukt om de verandering in het geheugen van de computer op te slaan.

Indien de regel OPSLAAN IN DE VIJFDE KOLOM moet worden veranderd in:

```
10 PRINT "OPSLAAN IN DE KOLOM"
```

moet het volgende worden gedaan:

- a Breng de cursor op de V van VIJFDE.
- b Hou de CONTROL-toets ingedrukt en druk op de DELETE-toets tot VIJFDE is verdwenen.
- c Druk op de RETURN-toets.

Indien u deze regel verder wilt wijzigen in:

```
10 PRINT "OPSLAAN IN DE TWEDE KOLOM"
```

moet u als volgt handelen:

- a Breng de cursor op de spatie tussen DE en KOLOM.

- b* Hou de CONTROL-toets ingedrukt en druk op de INSERT-toets tot er voldoende ruimte is ontstaan om het woord TWEEDE in te voegen.
- c* Type het woord TWEEDE.
- d* Druk op de RETURN-toets.

Type nu:

```
1 0 PRINT "SCHOOOL "  
2 0 PRINT "DOORN "  
2 0 0 PRINT "EINDE "
```

SCHOOOL met drie O's is te veel van het goede. Om deze fout te herstellen moet:

- a* de cursor op de L worden gezet;
- b* de DELETE-toets eenmaal worden ingedrukt;
- c* de RETURN-toets worden ingedrukt.

Regel 200 moet waarschijnlijk regel 30 zijn.

- a* Plaats de cursor op de 2 van 200;
- b* type het cijfer 3;
- c* druk gelijktijdig op de CONTROL- en DELETE-toets (om de 0 te verwijderen);
- d* druk op de RETURN-toets.

LIST het programma om te controleren of de wijzigingen juist zijn uitgevoerd.

Een regel verwijderen kan op twee manieren geschieden:

- 1 Type 10 en druk op de RETURN-toets.
- 2 Plaats de cursor in regel 20, druk gelijktijdig op een SHIFT- en op de DELETE-toets en druk op de RETURN-toets.

LIST nu het programma en alleen het EINDE is overgebleven.

Het invoegen van een regel hebben we in hoofdstuk 3 reeds besproken. Het is ook mogelijk een hele regel te vervangen.

Type:

```
1 0 PRINT "EEN REGEL VERVANGEN " ;  
2 0 PRINT "IS MOEILIJK "
```

Type nu:

```
2 0 PRINT "IS MAKKELIJK "
```

Als u het programma LIST, ziet u dat de eerste regel 20 vervangen is door de laatste regel.

Nu even een paar dingen op een rijtje zetten.

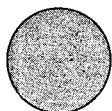
- 1 Het wissen van een teken of tekens gaat zo:
 - a* Als de cursor **rechts** van het teken staat: de DELETE-toets indrukken.
 - b* Als de cursor **op** het teken staat: de CONTROL- en DELETE-toets gelijktijdig indrukken.
 - c* Door spaties over de tekens te typen.

- 2 Het wissen van een regel gaat zo:
- Als de cursor op deze regel staat: een SHIFT- en de DELETE-toets gelijktijdig indrukken.
 - Door het regelnummer onder het programma te typen.

Taak

5a

Taak 5a afgerond



Laad het programma 'De rups' van de oefencassette. In dit programma staan een aantal fouten. Indien u de fouten ontdekt verander deze dan. Advies: LIST het programma en bekijk de regels. Regel 10 – 20 – 30 – 50 en 80 zijn goed.

Vindt u de oplossing niet, kijk dan in Aanhangsel A.

5.2

Het SAVEN van programma's

SAVEN is weer een woordje dat voor gebruik bij de computer eigenlijk niet te vertalen is. Het betekent 'veilig stellen', maar bij de computer is het het wegschrijven van programma's naar de cassetterecorder.

Nu u al iets weet van programma's maken is het nodig dat u leert op welke wijze de programma's kunnen worden bewaard op een cassettebandje.

In het vorige hoofdstuk is duidelijk geworden dat een programma niets anders is dan een lijst van instructies die in een bepaalde volgorde worden uitgevoerd door de computer. De juiste volgorde brengen we aan door iedere instructie een regelnummer te geven.

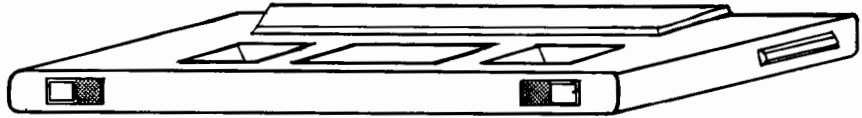
Er zijn twee manieren waarop een programma in de computer kan worden gebracht:

- 1 Getypt op het toetsenbord van de computer (zie hoofdstuk 4).
- 2 Geladen vanuit randapparatuur, zoals de cassetterecorder (zie hoofdstuk 1).

Zoals u wel gemerkt zult hebben, is het typen van een programma op het toetsenbord een tijdrovend karwei. Om nu te voorkomen dat hetzelfde programma – iedere keer dat het moet worden gebruikt – steeds weer moet worden getypt, is er een BASIC-commando – SAVE – dat het mogelijk maakt een programma naar een cassettebandje over te brengen.

Voor het overbrengen van een programma naar een cassettebandje kan iedere soort cassette worden gebruikt, bijvoorbeeld die waarop muziek wordt opgenomen. Het is aan te raden bandjes te gebruiken met een maximale speelduur van 30 minuten of minder. In de eerste plaats verkort het de zoektijd als er meerdere programma's op een bandje staan, en in de tweede plaats kunnen langere bandjes moeilijkheden in de recorder veroorzaken.

Het eerste dat van een cassette moet worden gecontroleerd, is of de beschermnokjes aan de bovenzijde nog aanwezig zijn.



De beschermnokjes kunnen worden verwijderd met een mesje of kleine schroevendraaier. Dit heeft tot gevolg dat er niets anders meer op de cassette kan worden opgenomen. Dit moet alleen worden toegepast als het een bandje betreft dat in geen geval mag worden gewist. Het is haast ondenkbaar dat u nu al de beschermnokjes wilt verwijderen. Als ze reeds verwijderd zijn kan dit worden hersteld door een klein stukje doorzichtig plakband over de gaatjes te plakken.

Wanneer u de juiste cassette heeft, plaatst u deze in de recorder en drukt u op de REWIND-knop om het bandje tot het begin terug te spoelen. Zet de teller aan de rechter bovenzijde van de recorder op 000. Spoel nu de band ongeveer 10 tellen voorwaarts door de ADVANCE-knop in te drukken. Dit is om er zeker van te zijn dat u op het magnetisch gedeelte van de band terecht komt.

Type nu een kort programma, tenzij u al een programma heeft dat u op de band wilt zetten.

Zet het geluid van de tv aan, type:

CSAVE

en druk op de RETURN-toets.

Twee pieptonen geven aan dat de PLAY- en RECORD-knoppen op de recorder gelijktijdig moeten worden ingedrukt.

Druk nogmaals op de RETURN-toets en de recorder begint met het opnemen van het programma op het cassettebandje.

Noteer, indien u het programma wilt bewaren, de begin- en eindstand van de teller op het lijstje bij het cassettebandje. Geef voor u zelf ook ieder programma een naam.

Indien u meerdere programma's op één zijde van het bandje opneemt, laat dan altijd, na de eindstand van het voorgaande programma, de cassette ongeveer 10 tellen doorlopen door middel van de ADVANCE-knop. Dit vergemakkelijkt later het terugzoeken van programma's bij het laden in de computer.

Behandel de cassettebandje met programma's zoals u muziekcassettes behandelt. Leg nooit een cassette op het televisietoestel.



Conditioes

In dit hoofdstuk worden de REM- en de IF... THEN-instructies behandeld. Het is de bedoeling dat de listing (tekst) van een programma altijd zo duidelijk mogelijk is.

Het is niet zo dat de maker van een programma ook altijd de programmeur is. U moet er naar streven programma's zo op te stellen dat een ander begrijpt wat de bedoeling van het programma is (gebruikersvriendelijk wordt dit genoemd). Een hulpmiddel hierbij is de REM-instructie.

6.1 De REM-instructie

REM is een afkorting van REMARK dat opmerking betekent. Deze instructie kan op iedere plaats in het programma worden gezet. De tekst die achter het woord REM wordt getypt zal door de computer niet in het geheugen worden opgenomen. De REM-instructie moet in een regel altijd ná de opdracht die moet worden uitgevoerd komen te staan.

Bijvoorbeeld:

```
10 PRINT "ZO WERKT DE REM-INSTRUCTIE" : R
EM DIT IS EEN VOORBEELD
20 REM DIT IS EEN VOORBEELD PRINT "ZO WE
RKT DE REM-INSTRUCTIE NIET"
```

Als u dit programma RUN't komt alleen de PRINT-instructie van regel 10 op het scherm. Doordat in regel 20 eerst het woord REM staat wordt de hele regel door de computer overgeslagen.

In regel 10 staat tussen de laatste aanhalingstekens en het woord REM een dubbele punt. Als deze dubbele punt wordt vergeten komt een ERROR-melding op het scherm te staan.

Als voorbeeld een programma met gebruikmaking van de REM-instructie. Type:

```
10 PRINT "\": REM MAAKT HET SCHERM SCHOO
N
20 REM OPTELLEN VAN GELIJKE GETALLEN
30 LET GETAL=1
40 PRINT GETAL ; "+" ; GETAL ; "-" ; GETAL+GETA
L: REM DEZE BEWERKING KOMT OP HET SCHERM
50 LET GETAL=GETAL+1
60 GOTO 40: REM DIT VEROORZAAKT DE LUS
```

Geen enkele maal de RETURN-toets vergeten?

RUN dit programma.

Stoppen door de BREAK-toets of stil zetten door de CONTROL- en 1-toetsen.

In regel 10 wordt in de REM-instructie een verklaring van het pijltje gegeven.

Regel 20 vertelt de bedoeling van het programma.

Regel 30 geeft de aanvangswaarde aan de variabele GETAL.

Regel 40 telt de waarden van getal + getal bij elkaar en drukt het resultaat af; de REM-instructie geeft aan dat alleen dit stukje op het scherm komt.

Regel 50 verhoogt bij iedere doorgang van de lus de waarde met 1.

Regel 60 brengt de lus terug naar 40; dit wordt verklaard in de REM-instructie.

De REM-instructie is in het bovenstaande programma natuurlijk onnodig veel gebruikt, maar dit was ook slechts een voorbeeld.

De bedoeling van de REM-instructie is om punten te verduidelijken, die door een programmeur (die niet de maker van het programma is) waarschijnlijk niet zullen worden begrepen.

6.2 De dubbele punt

De dubbele punt kan in ieder programma worden gebruikt en niet alleen bij een REM-instructie. Men kan er ook op één regel meerdere instructies mee plaatsen. Dit wordt gedaan om geheugenruimte te sparen of om een programma duidelijker te maken.

Type bijvoorbeeld:

```
10 REM DIT PROGRAMMA SLAAT REGELS OVER
20 PRINT "\":PRINT"REGEL EEN"
30 PRINT:PRINT:PRINT:PRINT"REGEL VIJF"
40 END
```

In regel 10 wordt in de REM-instructie de verklaring van het programma gegeven.

Regel 20 maakt het scherm schoon, de tweede instructie drukt de tekst af.

Regel 30 geeft drie lege regels en drukt daarna de tekst af van de vierde PRINT-instructie op die regel.

De lege PRINT-instructies kunnen ook worden vervangen door cursorbewegingen in een PRINT-instructie. Regel 30 zou dan als volgt luiden:

```
30 PRINT "↓↓↓ REGEL VIJF"
```

Welke van de twee manieren gebruikt wordt, hangt af van de opbouw van het programma.

Bij het gebruik van de dubbele punt moet goed worden gelet op twee instructies:

- 1 Na een REM-instructie kan geen instructie gescheiden door een dubbele punt worden gebruikt: alles na een REM-instructie wordt niet afgedrukt.
- 2 Na een GOTO-instructie kan geen instructie gescheiden door een dubbele punt worden gebruikt: de GOTO-instructie stuurt de computer naar de regel achter GOTO.

Dit kan dus niet:

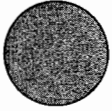
```
10 REM ZO MAAR EEN OPMERKING: PRINT"HAL
LO"
20 GOTO 10: PRINT"STOP MAAR"
```

dit wel:

```
10 PRINT"HALLO": REM ZO MAAR EEN OPMERK
ING
20 PRINT"STOP MAAR": GOTO 10
```

6a

Taak 6a afgerond



Maak een programma waarin de veelvouden van tien onder elkaar worden afgedrukt.

Vertel in een REM-instructie de bedoeling van het programma.

Het programma mag uit niet meer dan vier regels bestaan (dubbele punt gebruiken).

6.3

De IF...THEN-instructie

Tot nu toe hebben we gewerkt met de GOTO-instructie als we een lus in het programma wilden maken. Het nadeel van deze instructie is dat in het programma een eindeloze lus ontstaat. Willen we de lus kunnen controleren, dan moet gebruik worden gemaakt van een andere instructie, namelijk de IF...THEN-instructie.

IF...THEN betekent ALS...DAN. Dus als dit zo is dán gebeurt dat.

Een programma met een GOTO-instructie is alleen maar een programma met doe dit, doe dat. In een programma met de IF...THEN-instructie kan worden beslist wat in een bepaald geval moet gebeuren.

We zullen een programma opstellen om te telefoneren waarin de IF...THEN-instructie duidelijk wordt gemaakt.

- 1 Neem de hoorn van de haak, wacht op de kiestoon.
- 2 Draai het nummer.
- 3 ALS de hoorn wordt opgenomen, ga DAN naar 5.
- 4 Hoort u de in-gesprektoon: leg de hoorn neer en GA NAAR 1.
- 5 Spreek.
- 6 Leg hoorn neer; EINDE.

Bekijken we nu de regels:

In regel 4 ziet u de GOTO-instructie (GA NAAR), die een lus maakt naar regel 1 tot u de kiestoon hoort.

In regel 3 komt de IF...THEN (ALS...DAN)-instructie, de ontsnappingslus. ALS u de kiestoon hoort ga DAN naar regel 5.

De overige regels spreken voor zich zelf.

Regel 3 is anders dan de BASIC-instructies die we tot nu toe hebben geleerd: er is een **conditie** ofwel een **voorwaarde** ingebouwd. Zo'n instructie wordt een **voorwaardelijke** instructie genoemd.

De BASIC-instructies uit hoofdstuk 4 hebben een simpele vorm: **actie**.

Voorwaardelijke instructies hebben altijd de vorm: IF voorwaarde THEN actie.

In het telefoneervoorbeeld:

De **ACTIE** van het nummer draaien wordt alleen uitgevoerd indien aan de **VOORWAARDE** (het horen van de kiestoon) wordt voldaan.

De instructie op regel 3 veroorzaakt een lus die in het programma blijft tot men de kiestoon hoort en dus aan de voorwaarde in regel 3 is voldaan.

We zullen nu bekijken hoe voorwaardelijke instructies kunnen worden gebruikt in een zelf opgesteld programma.

Type, indien nodig, het commando `NEW` en type het volgende programma:

```
10 G=1
20 T=G*6
30 PRINT G; " * 6 = " ; T
40 G=G+1
50 GOTO 20
```

Het valt op dat in de regels 10, 20 en 30 de `LET`-instructie is weggelaten. Als we het tot nu toe geleerde in de praktijk hadden gebracht, zouden de regels er als volgt hebben uitgezien:

```
10 LET G=1
20 LET T=G*6
40 LET G=G+1
```

De `LET`-instructie mag worden weggelaten, want de computer leest voor `G=1` en `LET G=1` precies hetzelfde.

De reden dat we tot nu toe steeds de `LET`-instructie hebben gebruikt, is om duidelijk te maken dat de uitdrukking die na `LET` komt en gebruik maakt van het `=` teken, betekent: Maak de variabele ... gelijk aan ... Het betekent niet: De variabele ... is gelijk aan ... zoals in de gewone rekenkunde.

`G=G+1` is rekenkundig onmogelijk, maar in de `LET`-instructie staat: Maak `G` gelijk aan `G+1`, en dat is wel mogelijk.

Type nu `RUN` en druk op de `RETURN`-toets. Het resultaat is de tafel van 6 tot in het oneindige.

Nu kan de voorwaardelijke instructie worden toegepast om te bepalen hoe groot `G` mag worden.

Stop het programma door de `BREAK`-toets in te drukken en type de volgende regels:

```
25 IF G>10 THEN GOTO 60
60 END
```

Regel 25 bouwt de voorwaarde in: ALS `G` groter is dan 10 DAN ga naar 60. Regel 60 laat het programma eindigen.

`RUN` het programma en de tafel van 6 stopt bij `10*6`.

6.4 Relatietekens

In regel 25 komt u het volgende teken tegen: `>`. Dit is een relatieteken dat een relatie tussen twee waarden aangeeft.

`G>10` betekent dus: als `G` is groter dan 10.

Hierna volgt een overzicht van de relatietekens die in de `IF...THEN`-instructies kunnen worden gebruikt.

Teken	Betekenis	Voorbeeld
=	is gelijk aan	IF G=10 THEN ...
<	kleiner dan	IF G<10 THEN ...
<=	kleiner of gelijk aan	IF G<=10 THEN ...
>	groter dan	IF G>10 THEN ...
>=	groter of gelijk aan	IF G>=10 THEN ...
<>	ongelijk aan	IF G<>10 THEN ...

Taak

6b

Maak een aantal programma's, gebaseerd op de tafel van 6, waarin alle soorten relatietekens worden gebruikt.

Als voorbeeld:

```

10 G=1
20 T=G*6
30 IF G=10 THEN GOTO 70
40 PRINT G;" * 6=";T
50 G=G+1
60 GOTO 20
70 END

```

Probeer, alvorens de programma's te RUNnen, te bepalen waar het programma stopt.

Taak 6b afgerond



Er zijn een aantal kenmerken dat het eenvoudige programma van de tafel van zes gemeen heeft met ieder ander programma dat een lus en een voorwaardelijke uitgang bevat:

- 1 Omdat de waarde van G bepaalt of het programma al dan niet in de lus blijft, wordt G de **besturingsvariabele** genoemd.
- 2 De besturingsvariabele heeft altijd een **startwaarde**, in dit geval de waarde 1 ingebracht in regel 10.
- 3 Bij iedere lus wordt de besturingsvariabele met een bepaalde hoeveelheid verhoogd. In ons voorbeeld wordt G steeds met 1 vermeerderd, iedere keer dat regel 40 wordt doorlopen. De hoeveelheid die de besturingsvariabele iedere keer wordt verhoogd is de **vermeerdering**.
- 4 Uiteindelijk bereikt G de waarde 10. Nu is aan de voorwaarde in regel 25 voldaan, waardoor het programma de lus verlaat. Deze waarde is de **eindwaarde** van de besturingsvariabele.

Taak

6c

Lees het onderstaande programma en noteer de volgende punten:

- 1 De naam van de besturingsvariabele.
- 2 De startwaarde van de besturingsvariabele.
- 3 De vermeerdering waarmee de besturingsvariabele wordt verhoogd bij het passeren van de lus.
- 4 De eindwaarde van de besturingsvariabele.

```
10 PRINT "MUSEUM WISSELKOERSPROGRAMMA"
20 DOL=1
30 GUL=DOL*3.72
40 PRINT DOL;" DOLLAR=";"GUL;" GULDEN"
50 DOL=DOL+1
60 IF DOL>20 THEN END
70 GOTO 30
```

Taak 6c afgerond



Taak

6d

Taak 6d afgerond



Laad het programma 'Zeg het maar' van de oefencassette. Dit is weer een quizprogramma over de voorgaande hoofdstukken. Beantwoord de gestelde vragen.



Rekenen en INPUT-instructie

In de vorige hoofdstukken hebben we al verschillende malen met cijfers en getallen gewerkt. In dit hoofdstuk zullen we iets verder ingaan op het rekenen met de ATARI 800XL.

7.1 Bewerkingsvolgorde

We hebben gezien dat de computer voor diverse bewerkingen tekens gebruikt die afwijken van de bewerkingstekens die normaal bij het rekenen worden gebruikt. De schrijfwijze van deze tekens wordt hieronder samengevat.

Normaal	Op de computer
$10+5=$	$10+5=$
$10-5=$	$10-5=$
$10\times 5=$	$10*5=$
$10:5=$	$10/5=$
$10^5=$	$10^5=$

Optellen en aftrekken zijn dus wat tekens betreft gelijk aan de normale bewerkingstekens.

Voor het vermenigvuldigen wordt het teken * gebruikt om verwarring met de letter x te voorkomen.

Voor delen geldt hetzelfde; de : heeft voor de computer een andere betekenis, dus wordt het teken / als deeltteken gebruikt.

Het aangeven van machtsverheffen door een getal boven de lijn te schrijven, is voor de computer onmogelijk.

6 tot de macht 2 (6^2) is $6*6$

7 tot de macht 3 (7^3) is $7*7*7$

Op het toetsenbord staat een speciaal teken waarmee het machtsverheffen kan worden aangegeven. Dit teken is ^ en vindt u op de cursortoets →.

Door een SHIFT-toets en deze cursortoets gelijktijdig in te drukken, brengt u het machtsteken op het scherm.

Het getal vóór dit teken is het getal dat tot de macht moet worden verheven. Het getal ná dit teken wordt de exponent genoemd en geeft aan tot welke macht het eerste getal moet worden verheven.

2^4 wordt dus getypt als 2^4 en is gelijk aan: $2*2*2*2=16$

De volgorde waarin de bewerkingen door de computer worden uitgevoerd, is gelijk aan de volgorde in de normale rekenkunde:

Eerst machtsverheffen,
dan vermenigvuldigen of delen,
daarna optellen of aftrekken.

Als voorbeeld enige bewerkingen:

$7*7+1$ is dus eerst: $7*7=49$, en dan $49+1=50$.

Voor $1+7*7$ geldt dezelfde bewerking en geeft dus hetzelfde resultaat, want vermenigvuldigen gaat voor optellen.

Wat levert de volgende berekening op:

De uitkomst kan zijn: $7*6=42/3=14$
of: $6/3=2*7=14$

U ziet dat bij vermenigvuldiging en deling in één som de werkvolgorde niets uitmaakt.

De vastgestelde volgorde is echter: bij gelijkwaardige bewerkingen (zoals vermenigvuldigen en delen of optellen en aftrekken) wordt de volgorde door de som aangegeven.

7.2 Haakjes

We kunnen de computer echter dwingen van de normale werkvolgorde af te wijken door haakjes te gebruiken.

Indien de berekening:
 $2+4*4-3$ normaal wordt berekend, is de volgorde:
 $4*4=16+2=18-3=15$

Plaatsen we echter haakjes om delen van de berekening, dan wordt eerst de bewerking tussen de haakjes uitgevoerd en daarna de rest van de berekening.

$$(2+4)*(4-3)=6*1=6$$

Samengestelde berekeningen moeten voor de computer eerst worden herschreven in een, voor de computer begrijpbare formule. Zoals:

$$\frac{10+6+3^2}{8:4} \text{ wordt } (10+6+3^2)/(8/4)$$

Breuken worden door de computer allemaal herleid tot tiendelige breuken, met maximaal negen cijfers achter de komma. $1/7$ wordt dus 0.142857142

Taak

7a

Herschrijf, indien nodig, de bewerkingen en noteer het resultaat alvorens onderstaande berekeningen op de computer uit te voeren.

Berekening	In te voeren	Resultaat
$21+8-4$	PRINT	
$4 \times 8+2 \times 6$	PRINT	
$\frac{21+7 \times 2}{6 \times 3-8}$	PRINT	
12^2	PRINT	
$12^2 \times 4-16:4$	PRINT	
$2:3$	PRINT	

Taak 7a afgerond



De INPUT (invoer)-instructie maakt het mogelijk dat tijdens het lopen van een programma aan de programmeur gegevens worden gevraagd. Deze gegevens kunnen numerieke of string-variabelen zijn. Voor het invoeren van sting-variabelen moet eerst weer ruimte in het geheugen worden gereserveerd door middel van de DIM-instructie. De werkwijze van de INPUT-instructie zullen we aan de hand van een programma verduidelijken.

Type:

```
10 DIM STAD$(40)
20 PRINT "\ IN WELKE STAD WOONT U"
30 INPUT STAD$
40 PRINT "\ DUS U WOONT IN " ; STAD$
50 END
```

Het programma werkt als volgt:

10 reserveert geheugenruimte voor de stad (40 tekens is genoeg voor de langste plaats).

20 maakt het scherm schoon en stelt de vraag.

30 zet een vraagteken op het scherm, waarachter het antwoord moet worden getypt.

40 slaat een regel over (staat netter) en toont de tekst plus het antwoord van regel 30 op het scherm (vergeet niet de spatie tussen IN en het aanhalingsteken).

RUN het programma (na het beantwoorden van de vraag moet de RETURN-toets worden ingedrukt).

We kunnen het programma ook veranderen door bijvoorbeeld het huisadres en telefoonnummer te laten vragen. Verander het programma in:

```
10 DIM HUIS$(40), STAD$(40)
20 PRINT "\ WAT IS UW HUISADRES"
30 INPUT HUIS$
40 PRINT "\ IN WELKE STAD WOONT U"
50 INPUT STAD$
60 PRINT "\ WAT IS UW TELEFOONNUMMER"
70 INPUT TEL
80 PRINT "\ DUS U WOONT " ; PRINT ; PRINT HUI
S$
90 PRINT STAD$
100 PRINT "TEL " ; TEL
110 END
```

Zoals u ziet zijn in dit programma drie INPUT-instructies geplaatst.

Het is u waarschijnlijk opgevallen dat voor de TEL-variabele niet hoeft te worden gEDIMensioneerd. Een DIM-instructie is alleen nodig voor een string-variabele en niet voor een numerieke variabele.

Bij het intypen weer rekening houden met spaties op de juiste plaatsen.

In het programma is voor het overslaan van een regel afwisselend gewerkt met PRINT: en de cursorbeweging.

RUN het programma.

Het programma kan nog worden uitgebreid om meerdere malen een adres in te laten voeren.

Type nu:

```

10 DIM HUIS$(40),STAD$(40),VR$(3)
110 PRINT "WILT U VERDER GAAN? JA/NEE"
120 INPUT VR$
130 IF VR$="JA" THEN GOTO 20
140 IF VR$="NEE" THEN END

```

Regel 10 is uitgebreid met de DIM-instructie voor de vraag.
 Regel 110 wordt veranderd, hier komt de tekst van de vraag te staan.
 In regel 120 wordt het antwoord (JA of NEE) ingevoerd.
 Regel 130 stuurt, bij het antwoord JA, de lus naar 20.
 En regel 140 tenslotte beëindigt het programma als het antwoord NEE is.

Het stukje 110 tot en met 140 is vaker te gebruiken, bijvoorbeeld voor spelletjes waarin de gebruiker van het programma zelf moet beslissen of hij door wil gaan of stoppen.

Het is raadzaam om dit stukje en uit volgende hoofdstukken van dit boek meerdere programmadelen te noteren.

Ook delen van uw eigen programma's zijn dikwijls waard om bewaard te worden. U krijgt dan een bibliotheek met programmadelen waaruit geput kan worden bij het opstellen van programma's.

Waarschuwing

De GOTO-instructie mag nooit een lus maken naar de DIM-instructie. De computer probeert dan reeds geDIMensioneerde geheugenplaatsen opnieuw te dimensioneren, wat onherroepelijk tot een foutmelding leidt.

Dit kan dus niet:

```

10 DIM TEKST$(25)
20 TEKST$="GEEN GOTO NAAR DIM"
30 PRINT TEKST$
40 GOTO 10

```

Dit wel:

```

10 DIM TEKST$(25)
20 TEKST$="GOTO NAAR PRINT"
30 PRINT TEKST$
40 GOTO 30

```

Taak

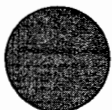
7b

Herschrijf het programma uit taak 6c zodanig dat:

- Er als titel WISSELKOERS DOLLARS komt te staan.
- Gevraagd wordt HOEVEEL DOLLARS WILT U WISSELEN?
- Via de INPUT-instructie het aantal om te wisselen dollars wordt ingevoerd.
- Het programma eindigt met de vraag VERDER GAAN JA/NEE.

Advies: Gebruik een lege PRINT-instructie of de cursorbeweging om ruimte tussen de regels te krijgen (bijvoorbeeld 30 PRINT:PRINT" ... of 30 PRINT"!..).

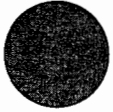
Taak 7b afgerond



7c

Laad het programma 'De lichtkrant' van de oefencassette. RUN het programma en voer de opdrachten uit.

Taak 7c afgerond



Geneste lussen en kleur

In vorige hoofdstukken hebben we gezien dat een lus een van de grondpijlers van een programma is. In dit hoofdstuk leren we een instructie waarmee tevoren kan worden vastgelegd hoeveel malen een lus moet worden doorlopen: de FOR ... NEXT-instructie.

In hoofdstuk 4 zagen we dat hoewel instructies normaal in de volgorde van de regelnummers worden uitgevoerd, het mogelijk is met de GOTO-instructie het programma te forceren door naar een andere regel over te springen en vandaaruit door te gaan.

Een voorbeeld is:

```
10 PRINT "DEZE ZIN WORDT STEEDS HERHAALD"
"
20 GOTO 10
```

We hebben ook gezien dat door het inbrengen van een besturingsvariabele bepaald kan worden hoeveel malen een lus wordt doorlopen. Zoals in:

```
10 A=1
20 PRINT "DEZE ZIN WORDT ";A;" KEER HERH
AALD"
30 A=A+1
40 IF A=10 THEN END
50 GOTO 20
```

8.1

De FOR ... NEXT-instructie

FOR ... NEXT wil zeggen VOOR ... VOLGENDE. Bij gebruik van de FOR ... NEXT-instructie kunnen de startwaarde, de vermeerdering en de eindwaarde van een variabele in een enkele instructie worden opgenomen. Het bovenstaande programma komt er dan als volgt uit te zien:

```
10 FOR A=1 TO 10
20 PRINT "DEZE ZIN WORDT ";A;" KEER HER
HAALD"
30 NEXT A
40 END
```

De startwaarde (1) en de eindwaarde (10) worden in regel 10 vastgesteld. Alles tussen regel 10 en regel 30 (NEXT) wordt het aantal malen van de waarde in regel 10 uitgevoerd.

In het bovenstaande voorbeeld wordt de vermeerdering automatisch op 1 ingesteld door de computer: bij iedere doorloop van de lus wordt A met 1 verhoogd.

Het is echter ook mogelijk een andere vermeerdering in te brengen door gebruik te maken van de STEP-vermelding (STEP==stap).

Type:

```
10 FOR X=0 TO 30 STEP 3
20 PRINT X
30 NEXT X
```

In dit programma wordt bij iedere lus de waarde met 3 verhoogd. Verlagen is ook mogelijk door het inbrengen van een negatieve STEP-waarde.

Type:

```

10 FOR N=50 TO 0 STEP -5
20 PRINT N
30 NEXT N

```

De waarde wordt nu steeds met 5 verminderd.

Taak

8a

Herschrijf onderstaand programma met gebruikmaking van de FOR...NEXT-instructie, in plaats van de IF...THEN- en de GOTO-instructie.

```

10 PRINT "\ "
20 G=1
30 PRINT G ; "+" ; G ; "=" ; G+G , G ; "*" ; G ; "=" ; G * G
40 G=G+1
50 IF G<21 THEN GOTO 30
60 END

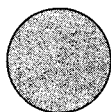
```

Tussen ;G+G en G; staat in regel 30 een komma. De komma zorgt ervoor dat het gedeelte achter deze komma in de tweede kolom wordt geprint, die op de 11e positie begint.

Het verschil tussen een ; en , is dus:

- a bij een puntkomma worden de opdrachten aan elkaar geprint;
- b bij een komma worden de opdrachten in kolommen van 10 posities geprint.

Taak 8a afgerond



8.2 De pauze-lus

De FOR...NEXT-instructie kan ook worden gebruikt om een pauze in een programma te brengen.

Wat zal er gebeuren in het onderstaande programma?

```

10 FOR PAUZE=1 TO 1000
20 NEXT PAUZE

```

De computer gaat 1000 keer in de lus 10-20-10 en er gebeurt niets.

Dit kost tijd en deze tijd is de pauze die de computer wacht.

Type nu:

```

10 PRINT "KLAAR . . . " ,
20 FOR PAUZE=1 TO 1000 : NEXT PAUZE
30 PRINT "AF!!!"

```

RUN dit programma.

In regel 10 wordt de tekst KLAAR... op het scherm geplaatst.

Regel 20 stelt een pauze in (NEXT kan door de dubbele punt te gebruiken op dezelfde regel worden geplaatst).

Regel 30 print AF!!! na de pauze.

Door de komma achter de tekst in regel 10 komt AF!!! in de volgende kolom achter KLAAR te staan.

Meerdere FOR...NEXT-lussen

Het is ook mogelijk twee of meer FOR...NEXT-lussen in een programma te gebruiken. Deze lussen mogen elkaar echter nooit snijden zoals in onderstaand schema duidelijk wordt gemaakt.

Toegestaan	Niet toegestaan
<pre> FOR A= FOR B= FOR C= NEXT C NEXT B NEXT A FOR A= FOR B= NEXT B FOR C= NEXT C NEXT A </pre>	<pre> FOR A= FOR B= FOR C= NEXT B NEXT C NEXT A </pre>

Een voorbeeld van meerdere FOR...NEXT-lussen geeft onderstaand programma dat de tafels van vermenigvuldiging op het scherm brengt. Type:

```

10 FOR TAFEL=1 TO 10
20 PRINT "\TAFEL VAN"; TAFEL; PRINT
30 FOR VERM=1 TO 10
40 PRINT VERM; " * "; TAFEL; " = "; VERM*TAFEL
50 NEXT VERM
60 FOR PAUZE=1 TO 2000: NEXT PAUZE
70 NEXT TAFEL
80 END

```

De werking van het programma per regel is als volgt:

- 10 De waarde van de tafel wordt op 1 gesteld en de vermeerdering wordt automatisch op 1 ingesteld.
- 20 Maakt het scherm schoon, print de tekst in reverse video en slaat een regel over.
- 30 Stelt de waarde van VERM (het getal waarmee wordt vermenigvuldigd) op 1 en stelt de vermeerdering automatisch op 1 in.
- 40 Drukt een regel van de tafel af.
- 50 Stuurt de lus terug naar 30 (10 x).
- 60 Geeft een vertraging als de tafel op het scherm staat.
- 70 Stuurt de lus terug naar 10 (10 x).

Taak

8b

Wijzig het onderstaande programma zodanig dat de regels STA... en STIL!!! achtmaal op het scherm worden afgedrukt. Maak een pauzelus (500) na STIL!!! en geef twee regels tussenruimte tussen de regels.

```

10 PRINT "STA...";
20 FOR PAUZE=1 TO 750: NEXT PAUZE
30 PRINT "STIL!!!"

```

Taak 8b afgerond

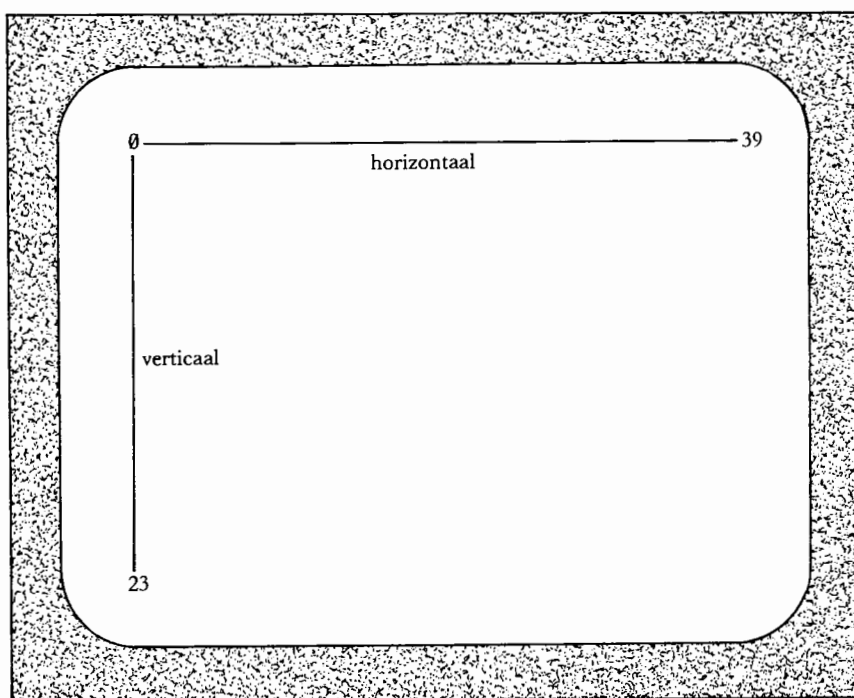
(Hier moet u geneste lussen gebruiken.)

Ook als u alleen met een zwart-wit televisietoestel werkt is dit gedeelte voor u van belang. Er staan verschillende programma's in die ook in zwart-wit leuke effecten geven.

Alvorens de kleurmogelijkheden van de ATARI 800XL te behandelen moet u eerst iets weten over de grafische mogelijkheden (GRAPHICS) van de computer.

De computer heeft 12 verschillende grafische mogelijkheden, genummerd van GRAPHICS 0 tot en met 11. Normaal staat de computer als hij wordt aangezet in de GRAPHICS 0-mode.

Een van de taken van de GRAPHICS-modes is het indelen van het scherm. In GRAPHICS 0-mode is het scherm horizontaal in 40 posities verdeeld en verticaal in 24. Deze posities zijn horizontaal genummerd van 0 tot en met 39 en verticaal van 0 tot en met 23 zoals op onderstaande tekening is aangegeven.



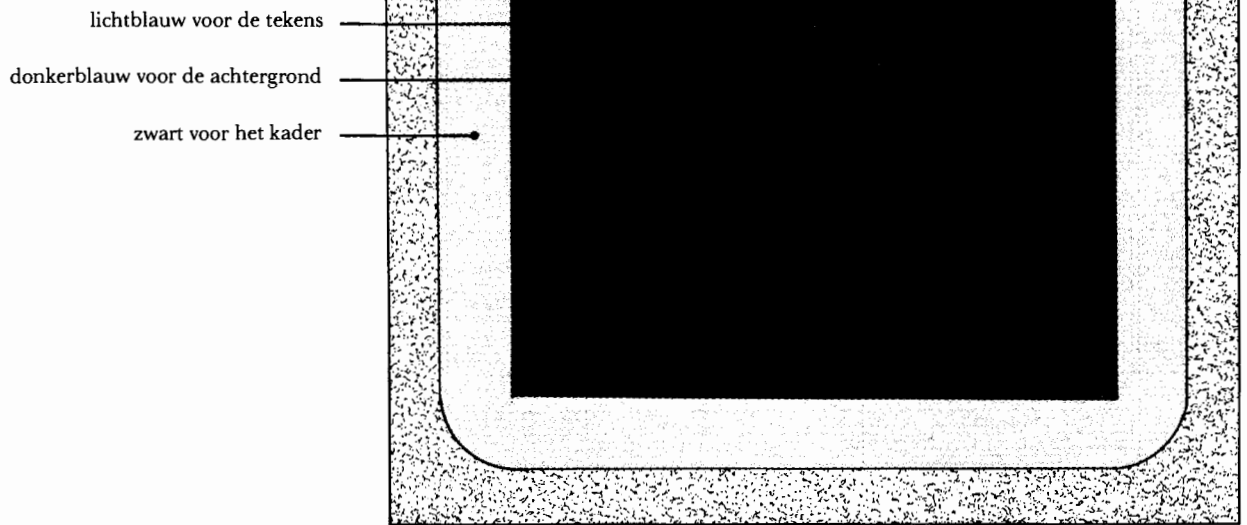
De GRAPHICS 0-mode is de zogenaamde tekstmode.

Alle andere grafische mogelijkheden worden uitvoerig besproken in hoofdstuk 17 tot en met 20.

De ATARI 800XL is een echte kleurencomputer. Niet minder dan zestien verschillende kleuren, ieder in acht schakeringen, staan tot uw beschikking. In totaal dus 128 verschillende kleurmogelijkheden.

Als de computer wordt aangezet, selecteert de computer bepaalde kleuren voor het scherm (de achtergrondkleur), voor de tekst (de tekenskleur) en voor de rand om het scherm (de kaderkleur).

Deze standaardkleuren zijn:



De basiskleuren waaruit u kunt kiezen zijn opgenomen in onderstaande tabel. Het cijfer voor de kleur moet worden gebruikt bij het instellen van de kleur.

0 grijs	8 blauw
1 goud	9 lichtblauw
2 oranje	10 turkoois
3 rood-oranje	11 groen-blauw
4 roze	12 groen
5 roze-paars	13 geel-groen
6 paars-blauw	14 oranje-groen
7 blauw	15 lichtoranje

8.5 De SETCOLOR-instructie

Voor het instellen van de kleuren wordt gebruik gemaakt van de SETCOLOR (kleur-instel)-instructie.

De SETCOLOR is zowel een instructie als een commando. Deze instructie kan dus zowel in een programma als in de directe mode worden gebruikt.

Achter het woord SETCOLOR worden drie getallen geplaatst. Deze geven (in GRAPHICS 0-mode) aan:

- a* achtergrond, tekst of kader,
- b* de kleurcode,
- c* de kleurhelderheid.

De codenummers voor de tekst, de achtergrond en het kader zijn:

- 1 voor de tekens,
- 2 voor de achtergrond,
- 4 voor het kader.

Het codenummer voor de kleur is gelijk aan het getal in de kleurtabel.

De codenummers voor de helderheid van de kleuren zijn:

0 - 2 - 4 - 6 - 8 - 10 - 12 - 14

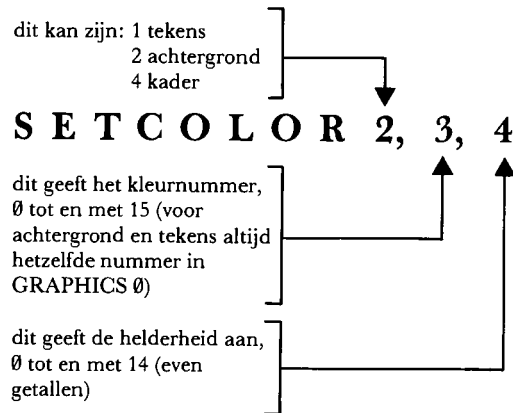
0 is de donkerste kleurschakering, 14 de helderste.

Zoals u ziet zijn de getallen die de helderheid bepalen alleen even getallen. Bij gebruik van een oneven getal wordt automatisch het lager gelegen even getal gekozen (dus 3=2, 9=8).

In de GRAPHICS 0-mode kunnen de kleuren van de tekens en de achtergrond niet afzonderlijk worden ingesteld. De achtergrondkleur bepaalt de tekstkleur, met dien verstande dat de tekstkleur een grotere helderheid heeft.

Indien u zelf de helderheid van beide instelt, kan dit ook worden omgedraaid, zoals we verderop in dit hoofdstuk zullen zien.

We zullen nu een voorbeeld geven voor het instellen van de kleur.



De instelling is dus het woord SETCOLOR - dan het cijfer 1, 2 of 4 - komma - een getal van 0 tot en met 15 - komma - een even getal van 0 tot en met 14.

De helderheidsinstelling van SETCOLOR1 en SETCOLOR2 mogen nooit gelijk zijn daar anders de tekst niet te lezen is.

Experimenteer een poosje met de kleurinstellingen. Geef verschillende instructies waarbij alle kleuren en helderheidsinstellingen aan bod komen.

Type bijvoorbeeld:

```
SETCOLOR1 , 1 , 0
SETCOLOR2 , 1 , 14
SETCOLOR4 , 1 , 10
```

Het onderstaande programma geeft een overzicht van de kleurmogelijkheden.

Type:

```
10 REM SCHERMKLEURINSTELLING
20 FOR KLEUR=0 TO 15
30 FOR HELDER=0 TO 14 STEP 2
```

```

40 SETCOLOR2,KLEUR,HELDER
50 PRINT"\SETCOLOR2,";KLEUR;" ";HELDER
60 FOR PAUZE=1 TO 500: NEXT PAUZE
70 NEXT HELDER
80 NEXT KLEUR
90 SETCOLOR 2,7,4: END

```

Het programma werkt als volgt:

Regel 20 stelt de variabele KLEUR op 0, de vermeerdering (1) en de eindwaarde worden bepaald.

Regel 30 stelt de variabele HELDER op 0, de vermeerdering (step 2) en de eindwaarde worden gegeven.

Regel 40 brengt de kleur en de helderheid in de SETCOLOR-instructie.

Regel 50 maakt het scherm schoon, print de tekst en de ingestelde kleur en helderheid.

Regel 60 is de pauze-lus.

Regel 70 stuurt de lus terug naar 30.

Regel 80 stuurt de lus terug naar 20.

Regel 90 brengt het scherm weer op de oude kleur (dit kan ook worden gedaan door de RESET-toets in te drukken).

Voor het instellen van de kaderkleur moeten de volgende regels in het programma worden gewijzigd:

```

40 SETCOLOR4,KLEUR,HELDER
50 PRINT"\SETCOLOR4,";KLEUR;" ";HELDER

```

Terwijl het programma loopt verdwijnt af en toe de tekst wanneer de helderheid van tekst en achtergrond gelijk zijn. Dit is leuk om in een programma te gebruiken.

Het volgende programma laat de tekst langzaam verdwijnen door de kleurhelderheid van achtergrond en tekst gelijk te maken.

Type:

NEW en type dan:

```

10 REM VERDWIJNENDE TEKST
20 FOR A=0 TO 14 STEP 2
30 SETCOLOR2,7,14
40 SETCOLOR1,7,A
50 PRINT:PRINT"\DEZE TEKST VERDWIJNT LA
NGZAAM"
60 FOR PAUZE=1 TO 500:NEXT PAUZE
70 NEXT A
80 SETCOLOR1,7,4

```

In regel 30 wordt de kleur van de achtergrond op de grootste helderheid gezet.

Regel 40 wijzigt de helderheid van de tekst bij iedere doorloop van de lus.

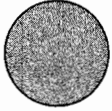
Regel 60 bepaalt de snelheid waarmee de tekst verdwijnt.

Regel 80 brengt de tekstkleur weer terug.

Taak

8c

Taak 8c afgerond

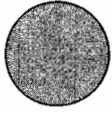


Maak nu een programma dat een eigen gemaakte tekst langzaam laat verschijnen.

Taak

8d

Taak 8d afgerond



Laad het programma 'Kleuren' van de oefencassette. RUN het programma en bekijk de kleuren in alle schakeringen.

DATA- en READ-instructies



In dit hoofdstuk leren we op welke manier een hoeveelheid gegevens (DATA) in een programma kan worden opgeslagen en hoe het programma deze informatie kan lezen.

9.1 De DATA-instructie

Door gebruik te maken van de DATA-instructie kan een programma aanmerkelijk worden verkort.

Indien in een programma bijvoorbeeld zes variabelen moeten worden gebruikt, zal dit programma, zonder gebruik van de DATA-instructie, er als volgt uitzien:

```
10 A=2
20 B=3
30 C=7
40 D=9
50 E=11
60 F=13
70 PRINT "7*11= "; C*E
```

enzovoort.

Al deze variabelen kunnen met gebruik van de DATA-instructie in één programmaregel worden geplaatst.

Type het onderstaande programma:

```
10 READ GETAL
20 IF GETAL=100 THEN GOTO 100
30 PRINT "\GEEF ANTWOORD OP DE VOLGENDE
VRAAG"
40 PRINT "↓HOEVEEL IS "; GETAL; " * "; GETAL;
50 INPUT VERM
60 IF VERM=GETAL*GETAL THEN PRINT "↓GOED
ZO!"
70 IF VERM<>GETAL*GETAL THEN PRINT "↓FOU
T!!"
80 FOR PAUZE=1 TO 1000: NEXT PAUZE
90 GOTO 10
100 END
110 DATA 2, 3, 7, 9, 11, 13, 100
```

Het programma vraagt u stuk voor stuk de uitkomst van de vermenigvuldiging te geven van de getallen in de DATA-regel.

In regel 10 ziet u de READ-instructie verschijnen. De READ-instructie leest het eerste getal in de DATA-regel en plaatst dit getal in het geheugen van de computer.

De DATA-regel kan op iedere plaats in het programma worden gezet, het regelnummer had dus ook 10, 30 of 60 kunnen zijn. De DATA-instructie wordt genegeerd totdat het programma de READ-instructie uitvoert.

Regel 20 is een bijzonder geval. Deze regel zet een zogenaamde **vlag** (in dit geval 100). Een vlag zorgt ervoor dat het programma wordt beëindigd. Zodra door de READ-instructie in de DATA-regel het getal 100 wordt gelezen, stopt regel 20 het programma.

Indien geen vlag in het programma staat komt de volgende foutmelding op het scherm:

```
ERROR- 6 AT LINE 10
```



Deze foutmelding geeft aan dat geen DATA-gegevens meer beschikbaar zijn.

Regel 80 geeft de pauze die nodig is om het antwoord te lezen.

Het is niet noodzakelijk alle DATA-gegevens op één regel te zetten. Vooral indien er veel DATA-gegevens in een programma komen te staan, is het overzichtelijker deze gegevens over meer regels te verdelen. Bijvoorbeeld:

```
10 DATA 128,148,12,32,64,3002,6040,16
20 DATA 221,346,585,689,1242,-1
30 READ N
40 IF N=-1 THEN END
50 PRINT (N*N)-(N+132)
60 GOTO 30
```

U moet altijd zorgen dat de vlag (in dit geval -1) als het laatste DATA-gegeven wordt ingevoerd. Indien -1 op regel 10 wordt geplaatst, zal regel 20 niet worden gelezen door de READ-opdracht.

In een READ-opdracht kunnen ook meerdere gegevens in één keer worden gelezen zoals uit onderstaand programma blijkt. Type:

```
10 PRINT "\ "
20 DATA 6,12,3,14,2,25,7,21,5,100,-6,-6
30 READ A,B
40 IF A=-6 THEN END
50 PRINT A;"*";B;"-";A*B
60 GOTO 30
```

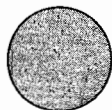
Taak

9a

Geef aan welke fouten in het onderstaande programma zitten. Verbeter deze fouten en RUN het programma.

```
10 READ GETAL
20 GETAL+3=6
30 PRINT RESULTAAT";3*GETAL
40 GOTO 20
50 DATA ,12,23,41,65,67
60 END
70 DATA 42,123,98,12,-3
```

Taak 9a afgerond



9.2

DATA-instructies met string-variabelen

In een DATA-instructie kunnen ook string-variabelen worden opgenomen. Zoals voor alle string-variabelen moet dan ook ruimte in het geheugen van de computer worden gereserveerd. Het aantal strings dat moet worden gelezen, moet in een DIM-instructie aan het begin van het programma worden opgenomen.

Hieronder volgt een quizprogramma dat u de namen vraagt van omroepen die bepaalde programma's verzorgen.

Type en RUN het volgende programma:

```

10 PRINT"\ "
20 DIM PGR$(25),OMR$(5),ANTW$(5)
30 READ PGR$,OMR$
40 IF PGR$="EINDE" THEN END
50 PRINT"welke omroep zendt ";PGR$;" uit"
60 INPUT ANTW$:PRINT
70 IF ANTW$<>OMR$ THEN PRINT "FOUT! het
is de ";OMR$
80 GOTO 30
90 DATA Zeg 'ns AAA,VARA,Televizier Maga
zine,AVRO,The A-Team,TROS,Bij Koos,VARA,
Nieuws,NOS
100 DATA Dynasty,VOO,Van Kooten en de Bie,
VPRO,Den Haag Vandaag,NOS,Puzzeluur,AVRO,
De Playback show,KRO,EINDE,EINDE

```

In de DIM-instructie zijn alle drie de variabelen opgenomen: programma (PGR\$), omroep (OMR\$) en antwoord (ANTW\$).

U ziet dat niet alleen in een PRINT-instructie kleine letters mogen worden gebruikt (weet u het nog: de CAPS-toets indrukken, voor hoofdletters SHIFT en CAPS), maar ook in de DATA-regels is dit toegestaan.

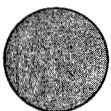
Regel 70 geeft aan dat het antwoord FOUT is indien het ingebrachte ANTW\$ niet gelijk is <> aan het door de READ-instructie gelezen antwoord.

In de DATA-regels zijn de PGR\$ en de OMR\$ twee aan twee naast elkaar opgenomen.

Taak

9b

Taak 9b afgerond



Wijzig het bovenstaande programma zodanig dat het een quiz wordt voor een door uzelf gekozen onderwerp, bijvoorbeeld films en hoofdrolspelers, liedjes en zangers.

In DATA-regels kunnen ook numerieke en string-variabelen door elkaar worden gebruikt. Er moet echter wel goed rekening worden gehouden met de plaatsing van de variabelen in de READ-instructie, zodat de juiste variabele wordt ingelezen.

Type:

```

10 PRINT"\ ": REM TWEE SOORTEN VARIABELE
N
20 DIM DIER$(10)
30 FOR AANTAL=1 TO 5

```

```

40 READ GETAL, DIER$
50 PRINT GETAL, DIER$
60 NEXT AANTAL
70 DATA 1, KAT, 2, HONDEN, 3, MUIZEN, 4, PAARD
EN, 5, VISSEN
80 END

```

U ziet dat in dit programma geen vlag is geplaatst, maar in regel 30 en 60 de FOR...NEXT-instructie wordt gebruikt. Deze methode kan altijd worden toegepast indien een vast aantal gegevens moet worden geprint. In de DATA-regel staan hier de numerieke en de string-variabelen om en om, omdat de READ-instructie deze leest in de volgorde: numeriek – string.

9.3 De RESTORE-instructie

RESTORE wil zoveel zeggen als opnieuw beginnen. Deze instructie kan in een programma met DATA-gegevens worden ingebouwd en zorgt ervoor dat het programma opnieuw wordt gestart indien het laatste DATA-gegeven is ingelezen.

De RESTORE-instructie wordt bijvoorbeeld in geluid- of muziekprogramma's gebruikt waarin dezelfde gegevens meerdere keren worden gelezen.

Hieronder een programma waarin de werking van de RESTORE-instructie duidelijk tot uiting komt. Type:

```

10 DATA 1, 2, 3, 4, 5
20 READ A, B, C
30 PRINT A, B, C
40 RESTORE
50 READ D, E
60 PRINT D, E

```

Het programma leest in regel 20 de eerste drie DATA-gegevens en print deze in regel 30.

De RESTORE-instructie in regel 40 brengt de READ-wijzer terug naar het eerste DATA-gegeven.

In regel 50 worden weer de eerste twee DATA-gegevens gelezen en daarna geprint in regel 60.

Taak

9c

Laad het programma 'Luchtballon' van de oefencassette. In dit programma wordt gebruik gemaakt van DATA-gegevens.

Nadat u met het programma gereed bent, moet de RESET-toets worden ingedrukt.

Type daarna:

```
LIST 20100, 20110
```

In regel 20100 staan twee woorden. (Deze twee woorden horen bij de instructie van het programma.)

In regel 20110 staan vijf woorden.

U kunt deze woorden vervangen door woorden die u zelf kiest, mits het aantal woorden gelijk blijft.

RUN daarna het programma opnieuw.

Indien u meer woorden wilt invullen in regel 20110 type dan:

```
LIST 10000,10110
```

In regel 10100 staat IF AW<5 THEN 640

Type nu: LIST 20110

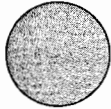
Vervang de woorden uit deze regel door het aantal woorden dat u wilt. Als regel 20110 te lang wordt, voegt u een nieuwe regel 20115 toe enzovoort.

Tel het aantal woorden en vervang in regel 10100

IF AW<5 THEN 640 door: IF AW<.. THEN 640

waarbij op de plaats van de puntjes een getal moet komen te staan dat gelijk is aan het aantal van de in de DATA-regel(s) getypte woorden.

Taak 9c afgerond



Geluid



In dit hoofdstuk worden de muzikale mogelijkheden van de ATARI 800XL behandeld.

U heeft al kennis gemaakt met het geluid en de muziek van de ATARI. Als eerste bij het testen van de AUDIO VISUAL in hoofdstuk 1 en verder bij het gelijktijdig indrukken van de CONTROL- en de 2-toets in hoofdstuk 2, wat een zoemtoon produceerde.

Deze zoemtoon kunt u ook opnemen in een PRINT-instructie van een programma om bijvoorbeeld de aandacht op een programmadeel te vestigen.

Het opnemen in een PRINT-instructie gaat als volgt:

Druk, na de aanhalingstekens achter PRINT, de ESC-toets in en daarna gelijktijdig de CONTROL- en de 2-toets. U krijgt dan tussen de aanhalingstekens hetzelfde pijltje op het scherm dat verschijnt bij het scherm schoonmaken (CLEAR) tussen de aanhalingstekens, maar nu in reverse video.

Dus een donker pijltje tegen een lichte achtergrond.

Dit teken zal in dit boek op de volgende wijze worden aangegeven: @.

Wilt u meerdere zoemtonen achter elkaar horen om de toon langer te laten duren, dan moet voor iedere toon de procedure worden herhaald (zoals bij de cursorbewegingen), dus ESC-toets - CONTROL- en 2-toets. Type en RUN het onderstaande programma (zet eerst het volume van de tv hoger).

```
1 0 PRINT "OPGELET@@@"
2 0 PRINT
3 0 PRINT "HEEFT U DE ZOEMTOON GEHOORD?"
```

10.1 Geluidsinstelling

Voor het produceren van geluid zijn verschillende gegevens nodig die moeten worden ingesteld en in een instructie moeten worden ondergebracht. Deze gegevens zijn:

- 1 Het geluidsregister.
- 2 De hoogte van de toon.
- 3 De helderheid van de toon.
- 4 Het volume van het geluid.

De ATARI beschikt over vier geluidsregisters. Deze zijn genummerd 0 - 1 - 2 - 3. De geluidsregisters kunnen afzonderlijk worden ingesteld. Er kunnen meerdere registers gelijktijdig in een programma worden ondergebracht voor het produceren van meerstemmige muziek.

De hoogte van de toon wordt ingesteld door een getal tussen 0 en 255. 1 is de hoogste toon en 255 de laagste.

De waarden voor een aantal toonhoogten zijn hieronder weergegeven.





De waarden van deze tonen (het getal dat moet worden ingesteld in de instructie) is:

C	D	E	F	G	A	B	C	D	E	F	G
121	108	96	91	81	72	64	60	53	47	45	40

Voor de tonen die tussen twee van deze tonen liggen, zoals A# of Bb, ligt de waarde tussen de twee getallen in:

A=72 B=64 dus A# of Bb=68

Een compleet overzicht van de beschikbare tonen is opgenomen in aanhangsel F.

Kleine afwijkingen aan de televisie of de computer kunnen de oorzaak zijn van een onzuivere klank van de ingestelde toon. Probeer dan de zuivere toon te verkrijgen door het getal van de waarde te veranderen. Is de A (72) bijvoorbeeld onzuiver, stel dan 71 of 73 in (een lager getal verhoogt de toon, een hoger getal verlaagt de toon).

Behalve de toonhoogte is ook de lengte van de noot (de duur) belangrijk. De noten worden in muziekstukken aangegeven op de wijze zoals in onderstaand schema staat.

hele	halve	kwart	achtste	hele	halve	kwart	achtste
met	met	met	met	met	met	met	met
noot	noot	noot	noot	stip	stip	stip	stip

120	60	30	15	180	90	45	23
-----	----	----	----	-----	----	----	----

Het getal onder de noot is de vertraging die in het programma moet worden gebracht om de noot de vereiste tijd te laten duren. Een punt achter een noot verhoogt de duur van de noot met de helft.

Evenals bij de toonhoogte kan ook de ingebrachte duur van computer tot computer verschillen. Indien nodig verlengt of verkort u de duur door de getallen evenredig te veranderen. Als 120 bijvoorbeeld 180 wordt, dan wordt 60 - 90.

De helderheid van de toon wordt bepaald door een getal van 0 tot en met 14, met 2 tegelijk oplopend (alleen even getallen dus). 10 is de helderste toon. Een lager of hoger getal geeft een vervorming van het geluid. Deze vervorming kan worden gebruikt om bepaalde geluidseffecten te verkrijgen.

Het volume van het geluid wordt bepaald door een getal van 0 tot en met 15. Bij 0 staat het volume af en bij 15 is de geluidsterkte het grootst. Indien meerdere geluidsregisters gelijktijdig worden ingesteld, mag de totale waarde van de volume-instelling nooit meer dan 32 bedragen.

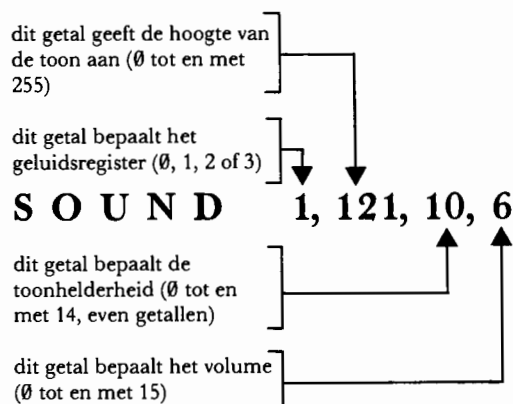
Bijvoorbeeld:
 3-stemmige volume-instelling: 10 - 14 - 8
 4-stemmige volume-instelling: 10 - 9 - 6 - 7
 of lager.

10.2 De SOUND-instructie

'Sound' betekent geluid. De SOUND-instructie zorgt ervoor dat de gegevens voor het ingestelde geluid in het geheugen van de computer worden opgeslagen.

De SOUND-instructie kan zowel in directe mode als in programma-mode worden gebruikt.

De algemene vorm van de SOUND-instructie wordt hieronder weergegeven.



De SOUND-instructie wordt dus in deze vorm op het scherm geplaatst of in een programma opgenomen.

10.3 Het programmeren van geluid

Voor het opstellen van geluidsprogramma's zijn een paar punten van belang. Door gebruik te maken van de FOR...NEXT-lussen kunnen de toonhoogte, de helderheid en het volume tijdens het lopen van het programma worden geregeld.

In dit boek zullen voor deze variabelen vaste namen worden aangehouden:

toonhoogte = TH

helderheid = HD

volume = V

De volumeknop van de televisie moet zodanig zijn ingesteld dat het geluid goed hoorbaar is.

In directe mode geprogrammeerd geluid blijft ingesteld totdat:

- a Het volume van de tv wordt afgezet,
- b De RESET-toets van de computer wordt ingedrukt.

Type:

SOUND1, 81, 10, 10

Denk erom dat de komma's op de juiste plaatsen moeten staan, anders krijgt u een ERROR-melding.

Als u de RETURN-toets heeft ingedrukt, hoort u de G totdat het geluid wordt afgezet of de RESET-toets wordt ingedrukt.
Stoppen door middel van de BREAK-toets is bij de SOUND-instructie onmogelijk.

Nu een programma dat u alle toonhoogten van de ATARI laat horen.
Type:

```
10 REM NOTEN VAN HOOG TOT LAAG
20 FOR TH=1 TO 255
30 SOUND 0, TH, 10, 10
40 FOR PAUZE=1 TO 25: NEXT PAUZE
50 NEXT TH
60 END
```

In regel 20 worden alle getallen tussen de 1 en 255 in de FOR...NEXT-lus gebracht.

In regel 30 wordt de SOUND-instructie ingesteld voor register 0, de toonhoogte voor TH, de toonzuiverheid op 10 en het volume op 10.

In regel 40 wordt een kleine pauze ingelast.

In regel 50 wordt de lus teruggestuurd.

In regel 60 wordt het programma beëindigd.

Het volgende programma vervormt het geluid door de toonhelderheid te variëren.

```
10 REM HELDERHEID VAN 0 TOT 14
20 FOR HD=0 TO 14 STEP 2
30 SOUND 1, 72, HD, 12
40 FOR PAUZE=1 TO 250: NEXT PAUZE
50 NEXT HD
60 SOUND 0, 0, 0, 0
```

Dit programma werkt op dezelfde wijze als het toonhoogteprogramma.

In regel 60 ziet u een manier om het geluid te beëindigen: in de SOUND-instructie worden alle instellingen op 0 gezet.

U heeft waarschijnlijk tweemaal een zuivere toon gehoord, toen de HD 10 of 14 was (10 geldt voor alle tonen, 14 voor de meeste).

Met het volgende programma gaat het volume van zacht naar hard. U krijgt hetzelfde effect als bij het verdraaien van de geluidsknop.

```
10 REM VOLUMEREGELING
20 FOR V=0 TO 15
30 SOUND 0, 60, 10, V
40 FOR PAUZE=1 TO 150: NEXT PAUZE
50 NEXT V
60 GOTO 20
```

Stop dit programma door de RESET-toets in te drukken.



10a

Speel een poosje met de geluidsmogelijkheden van de ATARI.
Maak een programma waarin zowel de toonhoogte als het volume gedurende het programma worden veranderd.
Advies: gebruik drie geneste lussen:

```
FOR TH=1 TO 255
FOR V=15 TO 0
FOR PAUZE=...
```

Taak 10a afgerond



Het volgende programma maakt gebruik van twee geluidsregisters, 0 en 1, die gelijktijdig werken. Het geproduceerde geluid lijkt op het telefoonsignaal.

```
5 REM TELEFOONSIGNAAL
10 FOR KEER=1 TO 7
20 SOUND 0,86,10,15: SOUND 1,88,10,15
30 FOR DUUR=1 TO 500: NEXT DUUR
40 SOUND 0,0,0,0: SOUND 1,0,0,0
50 FOR PAUZE=1 TO 1000: NEXT PAUZE
60 NEXT KEER
```

Regel 10 en 60 bepalen het aantal malen dat de bel overgaat.
Regel 20 stelt de beide SOUND-instructies in (gescheiden door dubbele punt).
Regel 30 bepaalt de tijdsduur van de bel.
Regel 40 stelt de beide SOUND-instructies op 0,0,0.
Regel 50 geeft een pauze tot de volgende bel.

In Aanhangsel F zijn verschillende programma's voor geluidseffecten opgenomen. Deze kunnen in bepaalde programma's worden verwerkt. Het is natuurlijk ook leuk om zelf programma's te maken, die geluidseffecten geven. Dit is eenvoudig te doen door in de bestaande programma's veranderingen in tijdsduur, toonhoogte, toonzuiverheid of volume aan te brengen.

Nu zullen we een programma opstellen voor het spelen van een eenvoudig melodietje. Voor dit wijsje gebruiken we register 1, verschillende noten, constante helderheid 10 en volume 10.

De melodie bestaat uit de noten:
C-C-G-G-A-A-F-F-E-E-D-D-C-G-G-F-F-E-E-D-G-G-F-F-E-E-D-C-C-G-G-A-A-G-F-F-E-E-D-D-C.

Voor deze noten moeten de waarden in de tabel worden opgezocht.

C=60
D=53
E=47
F=45
G=40
A=35

Deze waarden moeten in een DATA-regel worden ondergebracht in de volgorde van de melodie. Aan het einde zetten we de vlag op 0. Voor dit wijsje houden we de lengte (duur) van de noten gelijk: halve noten met stip. Dus de duur van iedere noot wordt 90. Type en RUN het onderstaande programma (vergeet niet het volume van de tv harder te zetten):

```

10 REM EENVOUDIG WIJSJE
20 READ TH
30 IF TH=0 THEN END
40 SOUND 1, TH, 10, 10
50 FOR DUUR=1 TO 90: NEXT DUUR
60 GOTO 20
70 DATA 60, 60, 40, 40, 35, 35, 45, 45, 47, 47, 5
3, 53, 60, 40, 40, 45, 45, 47, 47, 53
80 DATA 40, 40, 45, 45, 47, 47, 53, 60, 60, 40, 4
0, 35, 35, 40, 45, 45, 47, 47, 53, 53, 60, 0

```

U ziet dat de noot TH in de SOUND-instructie wordt ingelezen. Verder wijkt dit programma niet af van een normaal DATA-programma.

Als u dit programma kunt opstellen, geeft muziek op de ATARI voor u geen problemen meer.

Indien bijvoorbeeld de lengten van de noten moeten variëren, moet de variabele D (=duur) worden ingelezen en in de DATA-regel worden geplaatst.

Als voorbeeld:

Vier noten met een wisselende duur.

NOTEN: A=35 - F=45 - D=53 - C=60

DUUR van de noten: A=heel (120) - F=half (60) - D=half met punt (90) - C=kwart (30).

Hiervoor moeten de volgende regels van het programma worden gewijzigd en 80 verwijderd:

```

20 READ TH, D
50 FOR DUUR=1 TO D: NEXT DUUR
70 DATA 35, 120, 45, 60, 53, 90, 60, 30, 0

```

Eerst wordt de noot (TH) gelezen en in de SOUND-instructie geplaatst. Vervolgens wordt de duur (D) gelezen en in regel 50 geplaatst.

Wilt u ook nog het volume wisselend maken? Geen bezwaar.

Plaats in de READ-instructie de variabele V:

```
READ TH, V, D
```

Wijzig de SOUND-instructie in:

```
SOUND 1, TH, 10, V
```

en zet in de DATA-regel het volumegetal achter de noot (eerste noot bijvoorbeeld volume 6):

```
DATA 35, 6, 120
```

Wilt u twee of meer geluidsregisters inbrengen? Ook dit is niet moeilijk. Plaats de tweede SOUND-instructie achter de eerste in regel 40, gescheiden door een dubbele punt.

Breid dan de namen van de variabele uit, zodat ook de gegevens voor de tweede instructie in kunnen worden gelezen.

Het uitbreiden kan worden gedaan door andere letters te nemen of een 2 achter de bestaande letters te plaatsen.

Regel 40 wordt dan (uitgaande van de laatste methode):

```
40 SOUND 1,TH,10,V: SOUND 2,TH2,10,V2
```

De READ- en DATA-regels moeten natuurlijk ook worden aangepast. Hierin komen eerst de gegevens voor de eerste SOUND-instructie en dan de gegevens voor de tweede, vervolgens weer die voor de eerste, dan de tweede enzovoort:

```
20 READ TH,V,D,TH2,V2,D
70 DATA 35,6,120,47,10,30,53,8,60, enz.
          1e SOUND  2e SOUND  1e SOUND
```

Wordt dit een beetje te veel van het goede?

Oefen een poosje met makkelijke wijsjes en u zult zien dat het heel eenvoudig is. Maak desnoods zelf een programma waarbij in het ene SOUND-register lage tonen en in het andere de hoge tonen zijn opgenomen.

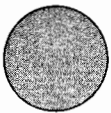
Breid dit programma steeds uit door eerst de duur van de noten te wijzigen en daarna een volumewisseling eraan toe te voegen.

In het verdere verloop van dit boek zal in verschillende hoofdstukken nog nader op geluid worden ingegaan.

Taak

10b

Taak 10b afgerond

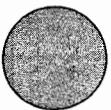


Laad het programma 'Geluiden' van de oefencassette. RUN dit programma en speel met de geluidsmogelijkheden. (Na het invullen van de gegevens achter het vraagteken moet de RETURN-toets worden ingedrukt.)

Taak

10c

Taak 10c afgerond



Laad het programma 'Herhaling' van de oefencassette. RUN het programma en beantwoord de vragen.

POSITION en beweging

In de voorgaande hoofdstukken zijn de meeste standaard BASIC-instructies en commando's behandeld. Vanaf dit hoofdstuk wordt uitgegaan van de veronderstelling dat u de stof van de eerste tien hoofdstukken beheerst.

In de programma's zullen ook enige veranderingen te zien zijn zoals het weglaten van de END-instructie, indien deze niet noodzakelijk is, het afkorten van variabelen tot dikwijls een enkele letter, het op één regel plaatsen van meerdere instructies gescheiden door een dubbele punt enzovoort.

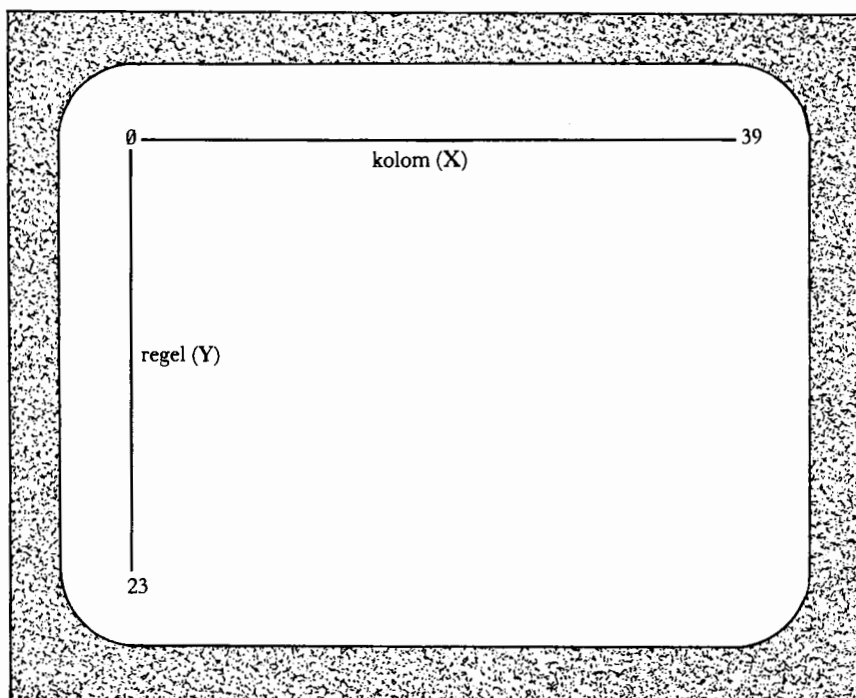
In dit hoofdstuk wordt de manier behandeld om tekst en grafische symbolen op diverse punten op het scherm te plaatsen en deze tekens, gedurende het lopen van het programma, te verplaatsen.

11.1 De POSITION-instructie

Alvorens deze instructie te bespreken zullen we eerst even naar het beeldscherm kijken.

In hoofdstuk 8 hebben we gezien dat in de GRAPHICS 0-mode het scherm horizontaal in 40 plaatsen (genummerd 0 tot en met 39) en verticaal in 24 plaatsen (van 0 tot en met 23) is verdeeld.

We kunnen ook zeggen dat het scherm horizontaal in 40 kolommen is verdeeld en verticaal in 24 regels.



De kolommen worden in variabelen met X aangeduid, de regels met Y. Normaal staat het eerste geprinte teken in de 2e kolom en de bovenste (0e) rij.

Willen we nu iets in het midden van het scherm plaatsen, dan kan dat worden gedaan door cursorbewegingen in de PRINT-instructie op te nemen, 20 naar rechts (X-as) en 12 naar beneden (Y-as).

Dit is natuurlijk een onmogelijke opgave en hier komt de POSITION (positie)-instructie van pas.

Als we in een programma het volgende typen:

dan zal het eerste teken uit de volgende PRINT-instructie in kolom 20 en rij 12 worden geplaatst.

De instructie ziet er dus als volgt uit: het woord POSITION – de horizontale kolom – een komma – de verticale rij.

We zullen nu gebruik maken van deze instructie om het woord MIDDEN in het midden van het scherm te plaatsen.

MIDDEN telt zes letters, dus de letter M moet drie plaatsen links van het midden (kolom 20) worden geplaatst. $20-3=17$, het eerste getal is dus 17. Het tweede getal is de middelste rij (12).

Type en RUN:

```
10 PRINT "\ "
20 POSITION 17, 12
30 PRINT "MIDDEN"
```

Dit is een eenvoudig programma.

Indien meerdere PRINT-instructies onder elkaar moeten worden geplaatst, moet voor iedere PRINT-instructie een POSITION-instructie worden gebruikt.

Type en RUN:

```
10 PRINT "\ "
20 POSITION 5, 5
30 PRINT "ATARI "
40 POSITION 5, 7
50 PRINT "COMPUTER"
```

Het is ook mogelijk de POSITION-instructie te gebruiken met een PRINT-instructie waarin de cursorbewegingen zijn opgenomen.

Zoals in:

```
10 PRINT "\ "
20 POSITION 34, 3
30 PRINT "█ (spatie spatie spatie) ↓←←← (spatie) 5 (spatie) ↓←←←
(spatie spatie spatie) █"
```

Dit programma geeft een licht gekleurd blok met in het midden het cijfer 5, aan de rechter bovenzijde van het scherm. Het blok wordt gevormd door spaties in reverse video af te drukken.

Wilt u dit resultaat bereiken zonder gebruik te maken van de cursorbewegingen, dan moeten drie POSITION-instructies worden gebruikt.

Het teken \ om het scherm schoon te maken, mag nooit in een PRINT-instructie na de POSITION-instructie worden geplaatst. De PRINT-instructie zorgt er dan voor dat het scherm schoon wordt gemaakt en print daarna de tekst links boven op het schone scherm.

De POSITION-instructie kan ook worden gebruikt om tekst over het scherm te laten bewegen, zoals in onderstaand programma duidelijk wordt gemaakt.

Type:

```

10 FOR Y=0 TO 23
20 PRINT "\ "
30 POSITION 14 , Y
40 PRINT "NAAR BENEDEN"
50 FOR P=1 TO 50: NEXT P
60 NEXT Y

```

In regel 10 wordt de aanvangs- en eindwaarde van Y bepaald (de rij in de POSITION-instructie).

Doordat in regel 20 het scherm schoon wordt gemaakt bij iedere doorloop van de lus 10-60, wordt ook steeds de tekst van regel 40 gewist.

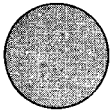
Regel 50 zorgt voor de pauze die nodig is om de tekst te kunnen lezen.

Taak

11a

Maak een programma waarin aan de rechterzijde van het scherm het woord OMHOOG van beneden naar boven gaat.

Taak 11a afgerond



Het is ook mogelijk in een programma twee teksten gelijktijdig over het scherm te laten bewegen in tegengestelde richtingen. Zoek dit zelf uit in de volgende taak.

Taak

11b

Laat gelijktijdig de tekst OMLAAG aan de linkerzijde van het scherm naar beneden, en de tekst OMHOOG aan de rechterzijde van het scherm naar boven bewegen.

Taak 11b afgerond



11.3 **Bewegende beelden**

Het is ook mogelijk bewegende beelden op het scherm te brengen. Dit effect wordt op ongeveer dezelfde wijze verkregen als de bewegingen in een tekenfilm. Hiertoe beeldt men óf verschillende tekens na elkaar op dezelfde plaats af, of beeldt een teken af, wist dit daarna en plaatst dan hetzelfde teken ernaast.

Hieronder volgt een programma dat een kloppend hart op het midden van het scherm plaatst. 91

Type en RUN dit programma:

```

5 PRINT "\ "
10 SETCOLOR 2, 3, 4
20 POSITION 19, 12
30 PRINT "■ |, ■"
40 FOR P=1 TO 100: NEXT P
50 POSITION 19, 12
60 PRINT "■ |T■"
70 FOR P=1 TO 200: NEXT P
80 GOTO 20

```

Regel 10 maakt de schermkleur rood.

Regel 20 en 50 zetten de af te beelden tekens op het midden van het scherm.

Regel 30 drukt het hartje in reverse video af. (CONTROL en ,).

Regel 40 en 70 zorgen voor de vertraging zodat het hartje lijkt te kloppen.

Regel 60 drukt het balletje in reverse video af. (CONTROL en T).

Stoppen door de BREAK-toets of de RESET-toets (RESET brengt ook de standaardkleur weer terug).

In dit programma kan ook geluid worden ingebracht zodat u het hart ook hoort kloppen.

Type de volgende regels en RUN het programma opnieuw:

```

31 FOR V=15 TO 0 STEP -1
32 SOUND 0, 250, 10, V
33 NEXT V

```

Een tweede manier om beelden te laten bewegen is het gebruik van spaties. In het onderstaande programma gaat de bal van links naar rechts over het scherm.

Type en RUN:

```

10 PRINT "\ "
20 FOR A=0 TO 39
30 POS. A, 5
40 PRINT "(spatie) |T"
50 FOR P=1 TO 20: NEXT P
60 NEXT A
70 GOTO 20

```

De spatie bedekt bij iedere doorloop van de FOR...NEXT-lus de afgebeelde bal; daarnaast wordt dan de volgende bal afgebeeld.

De pauze-lus is ingevoerd om een vertraging te geven; als u deze lus er uit laat zult u zien dat het haast onzichtbaar snel gaat.

11.4 Afkortingen

In regel 30 staat POS. in plaats van POSITION. Dit is de toegestane afkorting voor de POSITION-instructie.

Nu de programma's wat langer worden en u de meeste instructies en commando's kent, is het handig te weten welke in verkorte vorm in een programma mogen worden gebruikt.

Hieronder volgen de afkortingen van enige meest gebruikte instructies en commando's. Een volledige lijst is opgenomen bij de opsomming van alle instructies en commando's in Aanhangsel C.

Instructie	Afkorting
DATA	D.
DIM	DI.
FOR ... NEXT	F ... N.
GOTO	G.
INPUT	I.
LIST	L.
POSITION	POS.
PRINT	PR. of ?
SETCOLOR	SE.

Vergeet nooit de punt na de afkorting, want dan volgt onherroepelijk een ERROR-melding.

PRINT kan op twee manieren worden afgekort: PR. of het vraagteken zonder punt erachter. Het vraagteken is de meest handige afkorting: druk op een SHIFT-toets, hou deze ingedrukt en druk op het vraagteken en het aanhalingsteken.

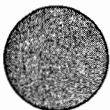
Bij nieuwe instructies of commando's zal vanaf dit hoofdstuk ook de afkorting direct worden gegeven.

In het verdere verloop van dit boek zullen volledige namen en afkortingen door elkaar worden gebruikt.

Taak

11c

Taak 11c afgerond



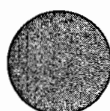
Maak een programma waarin het woordje WEG van POS. 1,1 naar POS. 23,23 loopt. WEG moet door spaties worden gewist.

Een vertraging van 50 invoegen tussen de bewegingen op het scherm.

Taak

11d

Taak 11d afgerond



Laad het programma 'Bewegen' van de oefencassette. Voer de in het programma gegeven opdrachten uit.

Getallen en functies

In hoofdstuk 7 hebben we al iets van de rekenmogelijkheden van de ATARI 800XL gezien. In dit hoofdstuk worden zeer grote getallen en enige functies behandeld.

Functies zijn bewerkingen op getallen of met variabelen. Het zijn bijvoorbeeld bewerkingen die vaak voorkomen en waarvoor een heel programma zou moeten worden opgesteld indien ze niet in ATARI BASIC zouden zijn opgenomen.

Er zijn twee soorten functies:

NUMERIEKE functies en STRING functies.

In dit hoofdstuk worden uitsluitend een aantal numerieke functies behandeld.

12.1 De E-notatie

Het grootste getal dat de computer direct op het scherm brengt is 9999999990 en het kleinste 0.000000001.

Wat gebeurt er indien met grotere of kleinere getallen moet worden gewerkt? Heel eenvoudig, dan zet de computer deze getallen in de E-notatie.

Type eens:

```
PRINT 2^35
```

De computer antwoordt met 3.4359643E+10

Dit is de E-notatie: het geeft een getal weer als een vermenigvuldiging met een macht van tien.

Het bovenstaande getal is: $3.4359643 \cdot 10^{10}$

Hoe groot is 10^{10} ? Zoals 10^2 gelijk is aan 100 (een 1 gevolgd door twee nullen), is 10^{10} gelijk aan 10000000000 (een 1 gevolgd door tien nullen) ofwel tien miljard.

Het voorbeeld 3.4359643E+10 betekent dus:

vermenigvuldig 3.4359643

met 10^E

tot de macht 10^{+10}

Het resultaat is: 34359643000

We kunnen zeggen dat de decimale punt zoveel plaatsen naar rechts opschuift als het getal achter E+ aangeeft.

Nu een getal 1.36284658E-6 ?

In dit geval schuift de decimale punt naar links en wel zes plaatsen.

1.36284658E-6 is dus 0.0000013284658.

12.2 De SQR-functie

De eerste functie die we zullen bespreken is de SQR-functie (SQR staat voor square root dat vierkantswortel betekent).

$\sqrt{25}$ is ook weer iets dat door de computer niet op het scherm kan worden gebracht.

De schrijfwijze van deze functie wordt voor de computer: SQR(X), waarbij X het getal is waaruit de wortel moet worden getrokken.

SQR kan in direct-mode en in programma-mode worden gebruikt.



SQR (25) -RETURN- geeft als antwoord 5.

10 PRINT SQR (0.0625) -RUN- geeft als antwoord 0.25.

Behalve de SQR-functie zijn in ATARI BASIC nog een heel stel wiskundige functies opgenomen. Voor degenen die geïnteresseerd zijn in wiskunde volgt hieronder een opsomming van enkele van deze functies.

Wiskundige notatie	Basic notatie	Betekenis
arctan(x)	ATN(X)	geeft de arctangens van X in radialen
cos(x)	COS(X)	geeft de cosinus van X in radialen
e ^x	EXP(X)	geeft de e-macht van X
ln(x)	LOG(X)	geeft de natuurlijke logaritme van X
sin(x)	SIN(X)	geeft de sinus van X in radialen
tan(x)	TAN(X)	geeft de tangens van X in radialen

x is hier het getal waarvan u de uitkomst wilt weten.

Voor een volledig overzicht van alle beschikbare reken- en wiskundige functies wordt verwezen naar het ATARI-handboek.

12.3 De INT-functie

Eerst iets over soorten getallen. We kennen hele getallen, zoals 1, 45, 12350 enz., en gebroken getallen, zoals 0.3, 0.476, 56.897 enz. De hele getallen worden ook wel de **integer**-getallen genoemd.

In BASIC hebben we een functie die een gebroken getal afrondt tot het daaronder liggende hele getal: de INT-functie (INT is integer).

De INT-functie wordt als volgt geschreven: INT(x), waarbij x het gebroken getal is.

Bijvoorbeeld:

```
PRINT INT ( 36 . 45 )      geeft 36
PRINT INT ( 3 . 6 )       geeft 3
```

U ziet dat de INT-functie alleen getallen naar beneden afrondt. Indien we getallen af willen ronden zoals dat veelal gebeurt – kleiner dan 0.5 naar beneden; groter of gelijk aan 0.5 naar boven – dan moet tussen de haakjes van de functie 0.5 worden bijgeteld.

```
PRINT INT ( 36 . 45 + 0 . 5 )      geeft 36
PRINT INT ( 3 . 6 + 0 . 5 )       geeft 4
```

Het gebruik van de INT-functie wordt in het verdere verloop van dit hoofdstuk nader besproken.

12.4 De RND-functie

Bij het ontwerpen van spelletjes komt het dikwijls voor dat u een willekeurig getal nodig heeft. Een willekeurig getal kan worden aangeroepen door de RND-functie. (RND is RANDOM en betekent willekeurig.) De functie geeft een willekeurig getal tussen 0 en 1.

Type en RUN het onderstaande programma:

```
10 FOR A=1 TO 15
20 PRINT RND(0)
30 NEXT A
```

U heeft nu 15 willekeurige getallen onder elkaar staan, allemaal kleiner dan 1 en met negen cijfers achter de komma.

Voor een spelletje heeft u niet veel aan zulke getallen. Stel dat u getallen wilt hebben tussen 1 en 10, wijzig regel 20 dan in:

```
20 PRINT 10*RND(0)
```

Nu heeft u 15 gebroken getallen tussen 1 en 10. om hier hele getallen van te maken komt de INT-functie van pas. Maak van regel 20:

```
20 PRINT INT(10*RND(0))
```

(Vergeet niet de dubbele haakjes achter de nul.)

Op het scherm staan nu willekeurige getallen van 0 tot en met 9.

Om daar getallen van 1 tot en met 10 van te maken moet bij ieder getal 1 worden opgeteld. Dus wijzigen we regel 20 in:

```
20 PRINT INT(10*RND(0)+1)
```

En daar is het gewenste resultaat.

U kunt door middel van de RND-functie, gecombineerd met de INT-functie, ieder willekeurig getal op het scherm brengen binnen de grenzen die u zelf bepaalt.

De algemene formule waarmee de grenzen worden bepaald is:

$$\text{getal} = \text{INT}((\text{bovengrens}-\text{ondergrens}) * \text{RND}(0) + \text{ondergrens})$$

Als voorbeeld: om een willekeurig getal te genereren tussen 15 en 25 wordt de formule:

$$\text{INT}((25 - 15) * \text{RND}(0) + 15) = \text{INT}(10 * \text{RND}(0) + 15)$$

Nu wat praktische toepassingen van de RND-functie.

Dobbelstenen werpen is een voorbeeld van het gebruik van de RND-functie. Iedere worp zal normaal een willekeurig getal opleveren van een tot en met zes, dus moet de functie ook deze getallen opleveren.

De functie wordt dan: $\text{INT}(6 * \text{RND}(0) + 1)$.

We zullen nu een programma opstellen waarin het werpen van een dobbelsteen wordt nagebootst.

Type en RUN:

```
10 ? "\n"
20 ? "DOBBELSTEEN WERPEN"
30 FOR A=1 TO 15
40 POS. 10, A+2
50 ? INT(6*RND(0)+1)
```

```
60 FOR P=1 TO 250: NEXT P
70 NEXT A
```

In regel 30 wordt het aantal malen dat de steen moet worden geworpen ingebracht.

Regel 40 zorgt ervoor dat de resultaten ongeveer midden onder de titel, onder elkaar, worden geprint.

Regel 50 is de functie.

Regel 60 last een pauze in om de worp te kunnen bekijken.

Een andere mogelijkheid voor het gebruik van de RND-functie is de computer zelf muziek te laten componeren.

Type en RUN het volgende programma:

```
5 REM PIANO MUZIEK
10 TH=INT(100*RND(0))
20 FOR V=15 TO 0 STEP-1
30 SOUND 0, TH, 10, V
35 FOR P=1 TO 10: NEXT P
40 NEXT V
50 GOTO 10
```

Stoppen door de RESET-toets.

Regel 10 genereert getallen tussen de 1 en 100.

Regel 20 stelt het volume in, van hard naar zacht (geeft een pianoklank).

In regel 30 wordt de noot uit regel 10 en het volume uit 20 in de SOUND-instructie gebracht.

12.5 De FRE-functie

FRE is FREE en betekent vrij. Door middel van deze functie kan het aantal vrije bytes worden opgevraagd. (Een byte is het stukje geheugen dat nodig is om een teken in het geheugen op te slaan.)

Bij lange programma's, en zeker bij programma's waarin veel met de tekenmogelijkheden wordt gewerkt, is het belangrijk dat u nog weet hoeveel geheugen beschikbaar is.

Type dan:

```
PRINT FRE(0)
```

en het aantal vrije bytes komt op het scherm te staan.

Taak

12a

Taak 12a afgerond



Laad het programma 'Hoger... lager...' van de oefencassette. Probeer het juiste getal te raden.

In dit programma wordt met de INT- en de RND-functie gewerkt.



POKE en PEEK

In dit hoofdstuk wordt de mogelijkheid besproken om in het vaste computergeheugen (ROM) veranderingen aan te brengen.

13.1 Het ROM-geheugen

Voor dit hoofdstuk is van belang op welke wijze in het geheugen wijzigingen kunnen worden aangebracht.

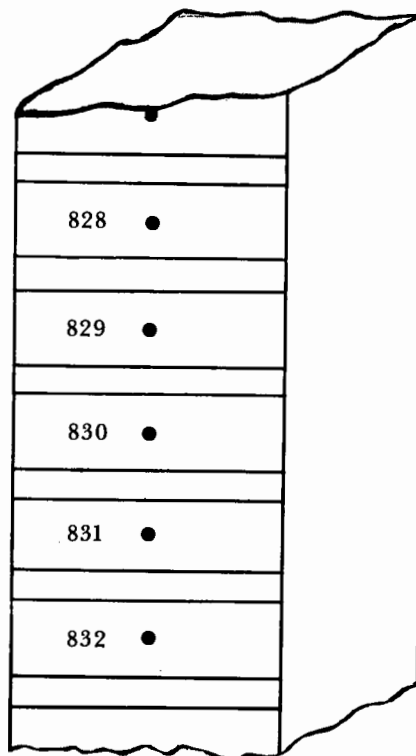
In hoofdstuk 3 hebben we gezien dat een deel van het geheugen (het RAM-geheugen) wordt gebruikt voor het opslaan van gegevens die door de programmeur in de computer worden gebracht.

Een ander deel van het computergeheugen is het ROM-geheugen. ROM staat voor Read Only Memory (alleen-lezen-geheugen). Dit beschrijft precies hoe dit geheugen werkt.

In het ROM-geheugen zijn alle gegevens opgeslagen die nodig zijn om de computer te laten werken: hoe letters, tekens en cijfers op het scherm moeten worden gebracht, de rekenfuncties, de BASIC-opdrachten, instructies hoe programma's vanaf cassetteband worden geladen in het geheugen en vanuit het geheugen op de cassetteband worden gebracht enzovoort.

Dit geheugen is, zoals alle computergeheugens, ingedeeld in geheugenplaatsen ofwel geheugenadressen. Ieder geheugenadres is – eenvoudig voorgesteld – een plaatsje in het geheugen, zoals ieder adres in een stad een vaste plaats in die stad is.

We kunnen het ROM-geheugen voorstellen als een kast met laden, die allemaal genummerd zijn, zoals op onderstaande tekening te zien is.



Een geheugenadres bestaat uit twee delen. Het eerste deel van het adres kunnen we met de straat vergelijken, het tweede deel met de bewoner (in dit geval een karakterteken, cijfer, letter, grafisch symbool en dergelijke). Het geheugenadres blijft bestaan, maar de bewoner kan veranderen. We zullen nu een adres gaan bekijken. 752,Ø zorgt ervoor dat het blokje dat de cursor aangeeft op het scherm komt. Als we nu de 'Ø' die achter de komma staat in een '1' veranderen (dus 752,1), verdwijnt de cursor.

Het veranderen van een geheugenadres wordt gedaan met behulp van de POKE-instructie. De POKE-instructie kan zowel in directe mode als in programma-mode worden gebruikt.

POKE is te vertalen met duwen. Door de POKE-instructie wordt iets in het geheugenadres geduwd waardoor dit adres verandert.

Type nu:

POKE 752,1

druk op de RETURN-toets

en de cursor is verdwenen.

Deze POKE-instructie is waard om te onthouden. Evenals het scherm schoonmaakteken (^) is deze instructie in de meeste programma's te gebruiken om de witte vlek van de cursor te laten verdwijnen.

Hoe krijgen we de cursor nu weer terug?

Heel eenvoudig door:

- a De oude toestand te herstellen en POKE 752, Ø te typen.
- b Door de computer uit te zetten.
- c Door de RESET-toets in te drukken.

Een adres in de POKE-instructie bestaat uit twee delen: het adres voor de komma en het getal achter de komma dat u zelf in kunt brengen.

Dit getal kan Ø tot en met 255 zijn. Dat wil niet zeggen dat elk van deze getallen een ander effect heeft indien het in een POKE-instructie wordt geplaatst, want het getal moet corresponderen met de mogelijkheden van het geheugenadres.

De cursor is bijvoorbeeld zichtbaar bij POKE 752, Ø en onzichtbaar bij POKE 752, (1 tot en met 255).

Door middel van de POKE-instructie kunnen instructies in een programma worden opgenomen, die met de normale programmeermethode een hoop programmaregels zouden vergen.

We zullen nu een programma maken waarin het voordeel van een POKE-instructie duidelijk blijkt.

De linker kantlijn van het scherm is opgenomen in geheugenadres 82, de rechter in 83. Indien we de waarde van deze kantlijnen gaan veranderen, wordt de tekst die daarna wordt geprint op een andere plaats afgedrukt.

Type en RUN:

```
1Ø POKE 752,1
2Ø POKE 82,1Ø: POKE 83,1Ø
3Ø PRINT "\VERTICAAL"
4Ø POKE 83,39
5Ø PRINT"HORIZONTAAL"
6Ø PRINT:PRINT"■alles is mogelijk■"
```

Regel 10 laat de cursor verdwijnen.

Regel 20 maakt de linker kantlijn (82) en de rechter kantlijn (83) gelijk, zodat maar één teken per regel kan worden geprint.

Regel 40 brengt de rechter kantlijn weer op zijn oude plaats.

Denk eens in hoeveel POS.-instructies of cursorbewegingen er nodig zijn om het woord 'VERTICAAL' ook verticaal op het scherm te brengen.

Ook alle SETCOLOR-mogelijkheden zijn in POKE-instructies onder te brengen.

Het geheugenadres voor de achtergrondkleur (schermkleur) is 710. Dat van de kaderkleur is 712.

De getallen die achter de komma in de POKE-instructie komen te staan zijn 0 tot en met 254, met twee gelijk oplopend (dus alleen even getallen). Het aantal getallen is dus 128 en dat is gelijk aan alle kleurmogelijkheden van de ATARI 800XL.

De 0 is kleur 0 (grijs) met helderheid 0.

254 is kleur 15 (lichtoranje) met helderheid 14.

Een volledig overzicht van de verhouding tussen SETCOLOR en POKE is te vinden in Aanhangsel E.

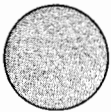
POKE 710, 254 is dus gelijk aan SETCOLOR 2, 15, 14

U ziet dat de POKE-instructie aanmerkelijk korter is dan de SETCOLOR-instructie. Het nadeel is echter dat een programma er niet duidelijker op wordt, tenzij er met REM's wordt gewerkt.

Taak

13a

Taak 13a afgerond



Stel een programma op, gebruik makend van de POKE-instructie, waardoor de achtergrond en het kader alle kleuren en helderheden in die kleuren doorlopen.

Breng tussen het wisselen van de kleurschakeringen een pauze in.

Wilt u een kleur even bekijken, druk dan gelijktijdig op de CONTROL- en de 1-toets (om door te gaan beide toetsen weer indrukken).

Het geheugenadres 755 kan voor verrassingen zorgen.

POKE 755, 0 drukt alles dat in reverse video staat normaal af en laat de cursor verdwijnen.

POKE 755, 1 wist alles dat in reverse video staat en laat de cursor verdwijnen.

POKE 755, 2 brengt de normale toestand terug.

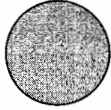
POKE 755, 3 laat de tekst uit de reverse video verdwijnen.

POKE 755, 4 drukt alle tekst ondersteboven af zonder cursor.

POKE 755, 6 drukt alle tekst ondersteboven af met cursor.

13b

Taak 13b afgerond



Maak een programma waarin de tekst 'VREEMDE ZAKEN' in reverse video wordt afgedrukt op POS. 13,8.

Laat vervolgens in een FOR...NEXT-lus POKE 755 alle mogelijkheden van 0 tot en met 6 doorlopen.

Maak een pauze van 750 na iedere doorloop van de lus.

(Advies: eerst scherm schoon, dan POS. en tekst, FOR, POKE, PAUZE, NEXT.)

Een handig geheugenadres is 54018. Door dit adres te gebruiken kan de cassette recorder aan en uit worden gezet. Indien u in de recorder een cassettebandje met muziek plaatst, kan door in dit adres verschillende getallen te POKE'n de recorder worden bediend.

Dit is ideaal indien u bij een eigen programma muziek of gesproken tekst wilt laten horen.

De PLAY-toets van de recorder moet natuurlijk zijn ingedrukt.

POKE 54018,52 schakelt de recorder in.

POKE 54018,60 schakelt de recorder uit.

In een programma kan dit op onderstaande wijze worden opgenomen:

```

10
.
.
100 POKE 54018,52
.
.
300 POKE 54018,60
.

```

Het programmadeel tussen de 100 en 300 wordt dan begeleid door muziek of gesproken tekst.

Waarschuwing

Indien u een onbekend geheugenadres met een willekeurig getal POKE't, kan het gebeuren dat het geheugen van de computer dol draait en tijdelijk onbruikbaar is. Zet dan de computer uit en na 5 seconden weer aan. Het geheugen is dan weer in de normale toestand.

Voorzichtigheid is geboden indien randapparatuur is aangesloten zoals een recorder of disc-unit, want u kunt dan onbedoeld programma's uitwissen.

13.3 De PEEK-functie

Het is ook mogelijk in een geheugenadres de opgeslagen waarde te bekijken zonder iets aan het adres te veranderen. Dit wordt gedaan door gebruik te maken van de PEEK-functie.

Type:

Na RETURN krijgt u: 0 op het scherm. Dat geeft aan dat het geheugenadres op dit moment 752,0 is en de cursor dus zichtbaar is. U kunt op deze wijze ieder adres bekijken.

Deze functie kan worden gebruikt om te zien of bepaalde toetsen worden ingedrukt.

Op het geheugenadres 53279 wordt een bepaalde waarde ingebracht, en wel bij het indrukken van:

- a de OPTION-toets waarde 3 (53279,3);
- b de SELECT-toets waarde 5 (53279,5);
- c de START-toets waarde 6 (53279,6).

Van dit adres en van de waarden kan gebruik worden gemaakt om een programma te selecteren of te starten.

Als in een programma de onderstaande regels worden opgenomen, kunt u zelf de start van het programma bepalen.

Type en RUN:

```

10 ? "\ DRUK OP DE START-TOETS "
20 IF PEEK(53279)=6 THEN GOTO 40
30 GOTO 20
40 POS.5,5: ? "HIER VOLGT HET PROGRAMMA"

```

Dit programma kan verder naar eigen idee worden aangevuld.

Regel 10 maakt het scherm schoon en print de tekst.

Regel 20 kijkt of de START-toets wordt ingedrukt (of dus het geheugenadres 53279,6 is); indien dit zo is GOTO 40. Indien de toets niet is ingedrukt zorgt:

Regel 30 voor een lus naar 20 tot de START-toets is ingedrukt.

Regel 40 zorgt voor de juiste positie en print de tekst.

Hetzelfde kan ook worden gedaan met de SELECT-toets om een bepaald gedeelte van een programma te selecteren.

De PEEK-functie kan ook in samenwerking met de POKE-instructie worden gebruikt.

Het geheugenadres 20 geeft de waarde van de ingebouwde tijdklok. Deze waarde is nauwkeuriger dan een waarde die we zelf inbrengen in een pauzelus. Deze tijdklok waarbij 60 eenheden ongeveer gelijk is aan 1 seconde, is ideaal voor de lengte van noten.

In een muziekprogramma moeten dan de regels 100, 110 en 120 van onderstaand programma worden opgenomen.

Type dit programma en vroeg in de DATA-regel eigen tonen in en de duur van deze tonen. Dus eerst toon en dan duur.

```

10 READ TH,D
20 SOUND 1,TH,10,12
100 POKE 20,0
110 T=PEEK(20)
120 IF T<D THEN GOTO 110
130 GOTO 10
200 DATA 53,80,60,90,212,120 enz.

```

In regel 10 wordt de toonhoogte en de duur ingelezen.
Regel 20 plaatst de toonhoogte in de SOUND-instructie.
Regel 100 start de klok op 0 en laat deze lopen.
Regel 110 stelt de waarde van de variabele T in op de klokwaarde.
Regel 120 vergelijkt deze waarde met de ingelezen D-waarde.
Regel 130 stuurt de lus terug naar 10.
Regel 200 geeft de DATA-gegevens voor TH en D.

Het programmadeel 100 tot en met 120 kan in ieder muziekprogramma worden opgenomen.

In Aanhangsel E zijn enige belangrijke POKE- en PEEK-adressen opgenomen.

Taak

13c

Laad het programma 'Weet u het' van de oefencassette. Dit is een quizprogramma over de laatste drie hoofdstukken.

Taak 13c afgerond



Array's

In dit hoofdstuk behandelen we het gebruik van een reeks (**array**) voor het invoeren van groepen numerieke variabelen.

14.1 Array's

In hoofdstuk 3 hebben we gezien dat als aan een variabele een waarde wordt toegekend, deze waarde in het geheugen van de computer wordt opgeslagen. De waarde wordt toegekend door bijvoorbeeld `A=60` te typen.

Als we in een programma niet aan één variabele een waarde willen toekennen, maar bijvoorbeeld aan 20 of desnoods aan 100, zullen er ellenlange reeksen variabelen komen te staan, zoals:

```
10 A=5
20 B=23
30 C=45
40 D=12
.
.
.
240 Z=145
```

Gelukkig kent ATARI-BASIC voor dit probleem een oplossing door de variabelen te DIMensioneren.

De DIM-instructie zijn we al tegengekomen bij de string-variabelen, waarbij voor het aantal letters ruimte in het geheugen werd gereserveerd. De DIM-instructie bij een array werkt iets anders: in plaats van ruimte te reserveren voor het aantal tekens, zoals bij de string-variabele, wordt nu ruimte gereserveerd voor het aantal numerieke variabelen.

Als een reeks geDIM ensioneerd wordt typt u:

```
DIM A(20)
```

Hiermee worden 20 geheugenplaatsen gereserveerd voor numerieke variabelen (of eigenlijk 21, want de nul doet ook mee). Deze variabelen hebben dan de namen `A0`, `A1`, `A2`, `A3` tot en met `A20`.

Iedere reeks in de DIM-instructie bestaat uit twee delen:

A eerst een letter of letters,
(20) dan een getal tussen haakjes dat het aantal variabelen aangeeft (dit getal wordt het subscript of bijschrift genoemd).

Aan de variabelen uit een reeks kunnen op de gewone wijze waarden worden toegekend. Dit toekennen van waarden kan op verschillende manieren gebeuren, hetgeen we met behulp van een paar voorbeelden duidelijk zullen maken.

Indien 30 variabelen moeten worden geDIM ensioneerd met waarden die gelijkmatig oplopen van 3 tot en met 90 moet het volgende programma worden opgesteld.

Type en RUN:

```
10 DIM G(30)
20 FOR W=1 TO 30
30 G(W)=W*3
40 NEXT W
```

Deze waarden zijn nu in het geheugen van de computer opgeslagen. U kunt dit controleren door te typen:

```
50 PRINT G(6); "+" ; G(12); "=" ; G(6)+G(12)
```

Type RUN en het antwoord moet zijn: $18+36=54$

Een tweede manier is het invoeren van willekeurige getallen in een DATA-regel.

Type en RUN:

```
10 DIM A(20)
20 FOR B=1 TO 20
30 READ C
40 A(B)=C
50 NEXT B
60 DATA 8,12,6,21,43,18,19,602,3,7,261,
51,97,110,81,73,65,14,15,1
```

Als test typt u:

```
PRINT A(3); "+" ; A(8); "=" ; A(3)+A(8)
```

Het antwoord moet zijn: $6+602=608$

De derde manier is het zelf invoeren van variabele waarden.

Type en RUN:

```
10 ? "\ "
20 DIM B(15)
30 FOR K=1 TO 15
40 ? "WELK GETAL WILT U INVOEREN ";
50 INPUT G
60 B(K)=G
70 NEXT K
80 ? "\ WELKE VARIABELE WILT U ZIEN ";
90 INPUT W
100 ? "DE WAARDE VAN B (" ; W ; ") = " ; B(W)
```

In regel 50 voert u 15 maal een waarde in.

In regel 90 geeft u aan van welke variabele de waarde moet worden getoond.

Als controle kunt u de waarden die u invoerde noteren en daarna vergelijken.

Het programma kan worden uitgebreid door de volgende regel toe te voegen:

```
110 FOR P=1 TO 1500: NEXT P: GOTO 80
```

Na een korte pauze komt dan steeds de vraag op het scherm:

WELKE VARIABELE WILT U ZIEN?

Om het praktische gebruik van de array te demonstreren geven we een programma waarin u zelf het aantal variabelen kunt invoeren.

Het programma houdt bij hoeveel maal de ogen 1, 2, 3, 4, 5 en 6 met een dobbelsteen worden geworpen indien het aantal worpen wordt ingevoerd.
Type en RUN:

```

10 ? "\HOEVEEL MAAL WERPEN" ;
20 INPUT A
30 DIM W(6)
40 FOR K=1 TO A
50 G=INT(6*RND(0))+1
60 W(G)=W(G)+1
70 NEXT K
80 ? : ? "WORP" , "AANTAL MALEN"
90 FOR C=1 TO 6
100 PRINT C , W(C)
110 NEXT C

```

Geef als INPUT 500 of 1000. Het antwoord laat even op zich wachten, zelfs een computer heeft tijd nodig om 500 maal een dobbelsteen te werpen.

In regel 20 worden de gewenste worpen ingevoerd, waarna regel 40 tot en met 70 de lus vormen voor het aantal worpen.

Regel 50 is de RANDOM-regel die voor het willekeurig werpen van de dobbelsteen zorgt.

Regel 60 is de gEDIMensioneerde regel.

Zonder de DIM-instructie zou deze regel er als volgt uitzien:

```

60 IF G=1 THEN W1=W1+1 : NEXT G
61 IF G=2 THEN W2=W2+1 : NEXT G
62 IF G=3 THEN W3=W3+1 : NEXT G
63 IF G=4 THEN W4=W4+1 : NEXT G
64 IF G=5 THEN W5=W5+1 : NEXT G
65 IF G=6 THEN W6=W6+1 : NEXT G

```

U ziet de besparing die de array geeft.

Taak

14a

Maak een programma waarin in de DIM-instructie het getal tussen de haakjes wordt ingevoerd (10 of meer).

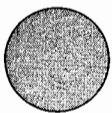
Geef aan iedere variabele een waarde die tweemaal zo groot is als het getal van de variabele (dus B(8)=16).

Laat de laatste programmaregel luiden:

```
PRINT " B ( 2 ) + B ( 8 ) = " ; B ( 2 ) + B ( 8 )
```

De uitkomst moet dan 20 zijn.

Taak 14a afgerond



14.2 Matrix

De ingewikkelder versie van een reeks heet **matrix**. In de matrix zijn er niet alleen rijen variabelen naar beneden, maar ook nog kolommen variabelen naar opzij.

We kunnen zeggen dat de matrix dubbel is geDIM ensioneerd.
De matrix heeft de onderstaande vorm:

A(1	,1	,2	,3	,4)
A(2	,1	,2	,3	,4)
A(3	,1	,2	,3	,4)
A(4	,1	,2	,3	,4)
A(5	,1	,2	,3	,4)

De matrix wordt op de volgende manier geDIM ensioneerd:

`DIM A (5 , 4)`

Het cijfer voor de komma geeft het aantal rijen aan, het cijfer achter de komma de kolommen. We kunnen zeggen dat de matrix tweedimensioneel is.

Het aantal variabelen dat op deze wijze in het geheugen van de computer wordt ingevoerd, is gelijk aan het produkt van de getallen tussen de haakjes. In dit geval dus $5*4=20$ variabelen.

De matrix werkt volgens dezelfde regels als de array:

- a Ze moet worden geDIM ensioneerd: `DIM B(6,3)`.
- b Er moet een waarde aan worden toegekend: `B(1,1)=8, B(1,2)=10` enzovoort.
- c De waarde wordt in het geheugen opgeslagen en kan worden gePRINT:

`PRINT B (5 , 2)`

Nu rijst natuurlijk de vraag wat voor nut de matrix heeft, omdat met een reeks al ontzettend veel variabelen in een DIM-instructie kunnen worden ondergebracht.

Als illustratie volgt hieronder een programma waarin de mogelijkheden van de matrix duidelijk worden getoond.

Type en RUN:

```

10 ? "\ "
20 DIM M ( 3 , 4 )
30 FOR E=1 TO 3
40 FOR T=1 TO 4
50 M ( E , T ) = 10 * E + T
60 POKE 82 , 0 REM KANTLIJN OP 0
70 PRINT M ( E , T ) ,
80 NEXT T : NEXT E

```

Regel 20 dimensioneert de matrix.

Regel 30 en 40 bevatten de FOR-instructies voor rijen en kolommen.

Regel 50 bepaalt de waarden van M(E,T).

Regel 60 zorgt voor een mooie schermverdeling (laat deze regel er eens uit en bekijk dan het resultaat).

Regel 70 print het resultaat.

Regel 80 maakt de lussen voor T en E.

De matrix kan worden gebruikt om de resultaten van een enquête te

noteren. Bij deze enquête zijn drie antwoorden mogelijk: 1 JA - 2 NEE - 3 GEEN MENING.

Als in het eerste getal achter de komma het aantal vragen wordt opgenomen, kan het tweede getal dienen voor het aantal te geven antwoorden.

Indien zes vragen aan twintig geënquêteerden worden gesteld, kan het resultaat er als volgt uitzien:

Vraag	Ja	Nee	Geen mening
1	12	3	5
2	4	16	0
3	0	17	3
4	18	0	2
5	14	6	0
6	8	10	2

Taak

14b

Noteer, alvorens het programma te typen en te RUNnen, het verwachte resultaat.

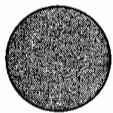
Wijzig het programma zodanig dat de waarde in de kolommen viermaal zo groot wordt.

```

10 ?"^\": DIM M(10,4)
20 POKE 82,0
25 ?"KOLOM 1", "KOLOM 2", "KOLOM 3", "KOLO
M 4"
26 ?"-----", "-----", "-----", "----
----"
30 FOR R=1 TO 10
40 FOR K=1 TO 4
50 M(R,K)=R*K
60 PRINT M(R,K),
70 NEXT K: NEXT R

```

Taak 14b afgerond

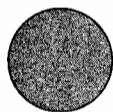


Taak

14c

Laad het programma 'Het gemiddelde' van de oefencassette. RUN het programma en voer de opdrachten uit.

Taak 14c afgerond



String-functies

In hoofdstuk 12 zijn al enige numerieke functies behandeld. In dit hoofdstuk komen **string-functies** aan de orde.

15.1 De ASCII-functie

Met behulp van deze functie kan ieder teken van de computer (letter, cijfer enzovoort) worden omgezet in een codegetal. Deze code heet de ASCII-code (American Standard Code of Information Interchange – de Amerikaanse standaardcode voor informatie-uitwisseling).

Deze code is voor het grootste gedeelte voor alle computers gelijk en kan worden gebruikt om onderling gegevens uit te wisselen tussen verschillende soorten computers.

Doordat de ATARI nogal wat extra tekens heeft, wordt de code voor de ATARI-computers ook wel ATASCII genoemd (ATARI ASCII).

De code die bij een teken hoort kan worden verkregen door de ASCII-functie in een PRINT-instructie te plaatsen.

Type:

```
PRINT ASC ( " A " )
```

Op het scherm verschijnt het getal 65, dat de code voor de hoofdletter A is (B=66, C=67 enzovoort).

De algemene vorm voor de ASCII-functie is: ASC("X"), waarbij X ieder teken van het toetsenbord kan zijn. Een volledige lijst van de ATASCII-code is opgenomen in Aanhangsel D.

De ASCII-functie kan onder andere worden gebruikt om woorden in een alfabetische volgorde te plaatsen.

15.2 De CHR\$-functie

De CHR\$ (karakterstring)-functie is het omgekeerde van de ASCII-functie. Met behulp van deze functie wordt op het scherm een teken geplaatst, dat wordt opgeroepen door de ASCII-code van dat teken in de CHR\$-functie te plaatsen.

Type:

```
PRINT CHR$ ( 65 )
```

Dit geeft de letter A (65 is de ASCII-code van de letter A).

Dit lijkt een onzinnige manier om iets op het scherm te krijgen, maar het wordt logisch als u weet dat sommige tekens niet direct op het scherm kunnen worden geprint.

Type:

```
PRINT " " TUSSEN AANHALINGSTEKENS " "
```

Zodra de RETURN-toets wordt ingedrukt komt een ERROR-melding op het scherm te staan. Tweemaal aanhalingstekens achter elkaar is onmogelijk.

Type nu:

```
PRINT CHR$ ( 34 ) ; " TUSSEN AANHALINGSTEKENS "
; CHR$ ( 34 )
```

En de tekst verschijnt keurig tussen aanhalingstekens op het scherm.
In totaal kunnen 256 getallen (van 0 tot en met 255) tussen de haakjes van de CHR\$()-functie worden geplaatst.

Indien u alle karakters op het scherm wilt bekijken, type en RUN:

```

10 POKE 82,0 : POKE 752,1 : REM KANTLIJN
NUL EN CURSOR WEG
20 FOR A= 0 TO 255 : REM ALLE MOGELIJKHE
DEN
30 PRINT CHR$(A) ,
40 FOR P=1 TO 150 : NEXT P : REM EEN KORT
E PAUZE TUSSEN DE KARAKTERS
50 NEXT A

```

Daar CHR\$(125) het scherm schoon maakt, vallen ongeveer halverwege alle geprinte tekens weg.

De tekens 128 tot en met 255 zijn de reverse video-tekens van de tekens 0 tot en met 127.

De CHR\$()-functie moet altijd buiten de aanhalingstekens van een print-instructie of zonder aanhalingstekens erin worden geplaatst. In de PRINT-instructie wordt de functie als een normale string behandeld en op het scherm als een tekst afgedrukt.

```
PRINT "CHR$( 125 )"
```

Het resultaat is dus: CHR\$(125)

De CHR\$()-functie kan ook voor spelletjes worden gebruikt. Als voorbeeld een programma dat met behulp van de RND-functie een doolhof op het scherm brengt.

Type en RUN:

```

10 PRINT CHR$( 125 ) : REM SCHERM SCHOON
20 PRINT CHR$( INT( 6.5+RND(0) ) ) ;
30 GOTO 20

```

Stoppen door middel van de BREAK-toets.

CHR\$(6) is het grafisch symbool /.

CHR\$(7) is het grafisch symbool \.

Door INT(6.5+RND(0)) als karakterstring te nemen, wordt de CHR\$()-functie willekeurig 6 of 7.

Als u meer \-strepen wilt, moet u het getal verhogen (b.v. 6.6); wilt u meer /-strepen, dan moet u het getal verlagen (b.v. 6.4).

In regel 10 staat de karakterstring voor het schoonmaken van het scherm: CHR\$(125). Dit is dus ook een manier om CLEAR in het programma op te nemen.

Zo zijn er nog ontzettend veel mogelijkheden met de CHR\$()-functie, zoals de cursorbewegingen CHR\$(28), (29), (30) en (31), de zoemer CHR\$(253). Een volledig overzicht van de CHR\$()-functie is opgenomen in Aanhangsel D.

Met de `LEN` (`LENGTH` is lengte)-functie kan het aantal tekens van een string worden bepaald.

Type en `RUN` het onderstaande programma:

```
10 DIM STD$(30) : REM STAD
20 PRINT "VOER EEN TEKST IN VAN TEN HOO
GSTE 30 TEKENS"
30 INPUT STD$
40 ? "\ " ; STD$ ; " HEEFT " ; LEN(STD$) ; " TEKE
NS IN ZIJN NAAM"
```

Regel 40 maakt eerst het scherm schoon, print dan de naam van de stad uit regel 30, telt het aantal tekens van de stad, print dit aantal en de rest van de tekst.

De `LEN`-functie telt alle tekens van de string, inclusief eventuele spaties. (Den Haag heeft dus acht tekens.)

De `LEN`-functie kan worden gebruikt om een onbekende tekst op het midden van het scherm te plaatsen.

Type en `RUN`:

```
10 DIM T$(30) : POKE 752,1
20 PRINT "VOER EEN TEKST IN VAN TEN HOO
GSTE 30 TEKENS"
30 INPUT T$
40 PRINT "\ "
50 POS. (40-LEN(T$))/2,6
60 PRINT T$
70 PRINT "↓↓links en rechts evenveel ruimte"
```

Regel 10 reserveert 30 tekens voor de string en verwijdert de cursor. Regel 50 zet de positie voor de kolommen op 40 min de helft van de stringlengte en, na de komma, op de zesde rij.

15.4 De GET-instructie

De `GET`-instructie neemt een enkel teken uit een apparaat en plaatst dit teken in een variabele. Alvorens dit uit te leggen gaan we eerst iets dieper in op de apparaten die hier worden bedoeld.

De apparaten worden aangeduid door een letter. Er zijn onder andere de volgende apparaten:

K = toetsenbord,
C = cassetterecorder,
S = beeldscherm,
F = beeldscherm en toetsenbord gecombineerd.

Deze apparaten kunnen voorbereid worden om buiten de normale taak nog iets anders te verrichten. De voorbereiding wordt gedaan door de `OPEN`-instructie (`OPEN` kan worden afgekort tot `O`).

Voor dit boek is alleen een `OPEN`-instructie naar het toetsenbord van belang. Deze instructie heeft de volgende vorm:

```
OPEN#1,4,0,"K:"
```

Het teken # staat boven de 3 (dus `SHIFT + 3` indrukken).

De OPEN-instructie heeft nu het toetsenbord (K) geopend voor het lezen (4) van een ingedrukte toets. (De andere getallen, 1 en 0, geven een code aan voor het in- en uitgaande controleblok en zijn nu verder niet van belang.)

De toets die gelezen moet worden wordt via de GET-instructie aangegeven. Deze instructie ziet er als volgt uit:

GET#1 , X waarbij X achter de komma de naam van een numerieke variabele is. Als waarde wordt aan deze variabele een getal toegekend dat gelijk is aan de ASCII-code van een ingedrukte toets. Dit kan iedere toets op het toetsenbord zijn.

20 GET#1 , WL – aan WL wordt nu een waarde toegekend gelijk aan de ingedrukte toets. Bij het indrukken van de A is de waarde: WL=65.

Een tweede mogelijkheid is zelf een waarde aan de variabele te geven die gelijk is aan een bepaalde toets en via een IF . . . THEN-instructie de GET-instructie alleen te laten werken indien de verlangde toets wordt ingedrukt.

```
40 GET#1 , G
50 IF G=65 THEN GOTO ...
```

Alleen wanneer nu de A(=65) wordt ingedrukt, zal de GOTO-instructie werken.

Hieronder volgt een programma waarin de eerste mogelijkheid wordt getoond. Type en RUN:

```
10 ?"^\": REM PIANOKLANKEN
20 OPEN#1 , 4 , 0 , "K : "
30 POS . 5 , 11
40 PR . "DRUK OP EEN WILLEKEURIGE TOETS"
50 GET#1 , TH
60 FOR V=15 TO 0 STEP -1
70 SOUND 1 , TH , 10 , V
80 NEXT V
90 GOTO 50
```

Het programma kan nog worden verlevendigd door de volgende regel toe te voegen:

```
75 POKE 710 , TH : REM KLEURVERANDERING
```

Regel 20 opent het toetsenbord voor het lezen van de ingedrukte toets. Regel 50 wacht tot een toets wordt ingedrukt, plaatst dan het ASCII-codegetal in de variabele TH.

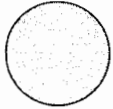
Regel 70: in de SOUND-instructie worden TH en V geplaatst, door de lus 80 – 60 neemt het volume af tot 0.

15a

Maak een programma waarin, door het indrukken van de toets T, de zoemer in werking treedt.

Verfraai het programma door midden op het scherm de tekst 'ZOEMERFEEST' te plaatsen en tevens het kader van kleur te laten veranderen.

Taak 15a afgerond



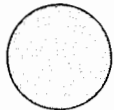
15b

Laad het programma 'De vlinder' van de oefencassette.

Indien u het programma RUnt, ontdekt u een aantal fouten. Probeer deze fouten uit het programma te halen.

Advies: LIST het programma. De regels 10 - 20 - 30 - 40 en 50 zijn goed. Kunt u de fouten niet vinden, kijk dan in Aanhangsel A.

Taak 15b afgerond



Programma's

In de voorgaande hoofdstukken hebben we steeds programma's gemaakt naar aanleiding van de stof in die hoofdstukken. In dit hoofdstuk wordt het opstellen van een uitgebreid programma behandeld. Tevens leren we hoe de besturing binnen een programma kan worden gecontroleerd door de GOSUB-instructie.

16.1 Het ontwerpen van een programma

Het belangrijkste bij het opstellen van een programma is eerst een ontwerp op papier te maken. Het is verleidelijk gelijk met het intypen van een programma op de computer te beginnen, maar op de lange duur blijkt dat dit veel meer tijd kost dan eerst een schema op papier te maken.

Zoals bij ieder probleem dat moet worden opgelost, is de eenvoudigste methode de zaak in kleine stukken te verdelen en deze later tot een geheel samen te voegen.

Als dit op de juiste wijze wordt gedaan, kan ieder stukje van het probleem als een programma worden behandeld en uitgetoetst. Vervolgens kan een supervisieprogramma worden gemaakt dat alle programmadelen tot één geheel samenvoegt.

Zo'n supervisieprogramma wordt de **supervisor** genoemd. De supervisor zorgt dat alle programmadelen in de juiste volgorde worden uitgevoerd.

In ATARI BASIC bestaat een instructie waarmee van de supervisor naar de verschillende programmadelen kan worden gesprongen en van die delen weer terug naar de supervisor. Deze instructie is de GOSUB-instructie.

16.2 De GOSUB-instructie

We hebben gezien dat normaal een programma wordt uitgevoerd in de volgorde van de regelnummers. De enige uitzondering hierop is de GOTO-instructie, die de besturing van het programma naar een regel stuurt welke in de GOTO-instructie is aangegeven.

Met de GOSUB (afgekort GOS.) kunnen we de **besturingsstroom** naar een serie **subroutines** (programmadelen) sturen, deze subroutines uit laten voeren en de besturingsstroom weer terugsturen naar het hoofdprogrammadeel, de supervisor.

Alvorens zo'n programma samen te stellen zullen we eerst bekijken hoe de GOSUB-instructie werkt. Type en RUN:

```

10 PRINT "\ "
20 PRINT "DIT IS EEN GOSUB DEMONSTRATIEPROGRAMMA "
30 GOSUB 100
40 PRINT "\DIT WAS SUBROUTINE 1":FOR P=1
TO 400:NEXT P
50 GOSUB 200
60 PRINT "DIT WAS SUBROUTINE 2":FOR P=1
TO 400:NEXT P
70 GOSUB 300
80 PRINT "DIT WAS SUBROUTINE 3"
90 END
100 POS. 0,10

```



```

110 PRINT "DEZE SPRONG WORDT GEMAAKT DOOR
    REGEL 30"
120 FOR P=1 TO 750:NEXT P: REM PAUZE OM
    DE TEKST EVEN TE LATEN STAAN
130 RETURN :REM TERUG NAAR 40
200 POS. 18, 10
210 ? "\ (3 spaties) ↓ ← ← ← (spatie) 2 (spatie) ↓ ← ← ← (3 spaties) █"
220 POKE 710, 20 :REM KLEUR GOUD
230 FOR P=1 TO 50: NEXT P
240 RETURN
300 POKE 710, 148: REM TERUG NAAR NORMALE
    KLEUR
310 PRINT "\EINDE PROGRAMMA ↓ ↓"
320 RETURN

```

In de regels 130, 240 en 320 staat het woord RETURN (terug). Dit is de RETURN-instructie die bij iedere GOSUB-instructie hoort. De RETURN-instructie zorgt ervoor dat, nadat het deelprogramma (de subroutine) is uitgevoerd, de besturingsstroom teruggaat naar de regel die volgt op de laatste GOSUB-instructie die is uitgevoerd.

Dus:

```

30 = GOSUB 100           130 = RETURN naar 40
50 = GOSUB 200           240 = RETURN naar 60
70 = GOSUB 300           320 = RETURN naar 80

```

Als u dit onder de knie heeft is er geen enkel probleem om ellenlange programma's op te stellen. De GOSUB kan ook handig worden gebruikt om meerdere malen naar eenzelfde subroutine te sturen.

```

10 ...
20 GOSUB 300           in dit programma wordt
30 ...               tweemaal regel 300
40 GOSUB 400           aangestuurd
50 GOSUB 300
60 END

```

16.3 Eisen waaraan een programma moet voldoen

Voor het opstellen van een goed lopend programma moeten enige punten in het oog worden gehouden:

- 1 Het programma moet duidelijk zijn, zodat iemand die het programma intypt – die niet de maker van het programma is – toch het programma volkomen begrijpt. Een oplossing is REM-instructies te gebruiken bij onduidelijke regels.
- 2 Een programma moet een begin en een einde hebben. Als begin kan bijvoorbeeld een titel worden gebruikt, als einde de mededeling 'EINDE PROGRAMMA' of iets dergelijks.
- 3 Tijdens het lopen van een programma moeten, indien nodig, duidelijke aanwijzingen worden gegeven zoals: druk op de RETURN-toets – spatiebalk gebruiken of iets dergelijks.
- 4 Het lopende programma moet zo duidelijk zijn dat een gebruiker, die het programma van een cassettebandje laadt, geen mondelinge uitleg nodig heeft.

We zullen nu een programma opstellen waarbij gebruik wordt gemaakt van een supervisor en subroutines. Het programma is een aanbeveling voor het gebruik van melk (u kunt natuurlijk ook een andere favoriete drank invullen).

In de eerste plaats hebben we een titel nodig voor het programma. Daar nemen we het woord MELK in grote letters voor, die zijn samengesteld uit *-tekens.

Ten tweede de aanbeveling: MINSTENS IEDERE DAG EEN GLAS, daarbij een glas dat wordt gevuld en geleegd.

Dan de constatering MELK DE WITTE MOTOR. Met deze zin vullen we gedeeltelijk het scherm.

Als einde van het programma laten we de titel weer zien die langzaam vervaagt.

Het programma vullen we op met een tussenstukje dat tweemaal wordt gebruikt. De woorden MELK MOET laten we links omlaag en rechts omhoog over het scherm lopen in een andere kleur.

Hieronder volgen de subroutines. Neem altijd voldoende ruimte voor de nummering van de regels. Begin een programma met 100 en laat de regels met 10 oplopen, zodat er voldoende ruimte is om overal extra regels in te voegen.

Het eerste deel geeft in REM-instructies aan waar het programma over gaat en laat de cursor verdwijnen.

Na iedere subroutine even het RUN-commando geven om te zien of dit deel goed loopt. Dan de RESET-toets indrukken om op een schoon scherm de volgende subroutine te typen.

Type:

```
100 REM -----
110 REM      MELK
120 REM -----
130 POKE 752,1 REM CURSOR WEG
```

Hierna moet de supervisor komen, die we echter als laatste typen wanneer de rest van het programma bekend is.

Nu de titel in *-tekens met een witte lijn eronder. De lijnen worden gemaakt door spaties in reverse video (23 spaties).

```
1000 REM NAAM VAN HET PROGRAMMA
1010 ?"\" : POS. 8, 8
1020 POKE 82,8: REM KANTLIJN VAST OP KOL
OM 8
1030 ?" * * * * * * * * * * "
1040 ?" ** * * * * * * * * "
1050 ?" * * * * * * * * * "
1060 ?" * * * * * * * * * "
1070 ?" * * * * * * * * * "
1080 ?"↓↓↓↓" REM VIER REGELS LAGER
1090 FOR A=1 TO 2
1100 ?"█(23 spaties)█": REM WITTE LIJN
1110 ?"↓↓↓": NEXT A
1120 FOR P=1 TO 600: NEXT P: REM TIJD DA
T MELK OP HET SCHERM BLIJFT
1130 RETURN
```


ATARI
MUSEUM

```

4050 FOR P=1 TO 40: NEXT P
4060 NEXT A
4070 FOR P=1 TO 500: NEXT P
4080 RETURN

```

De eindroutine besluit het programma. De tekst verdwijnt langzaam uit beeld.

Type:

```

5000 REM VERVAGEN TITEL
5010 FOR A=1 TO 14
5020 SETCOLOR 2,7,14
5030 SETCOLOR 1,7,A
5040 FOR P=1 TO 100: NEXT P
5050 NEXT A
5060 RETURN

```

Nu moet alleen de supervisor nog worden opgesteld. Daarin moeten we zorgen dat alle subroutines op de juiste wijze worden aangestuurd, zodat het programma in de juiste volgorde verloopt.

```

200 REM SUPERVISOR
210 GOSUB 1000: REM TITEL
220 GOSUB 2000: REM TUSSENGEDEELTE
230 GOSUB 3000: REM EEN GLAS PER DAG
240 GOSUB 4000: REM WITTE MOTOR
250 GOSUB 2000: REM TUSSENGEDEELTE
260 GOSUB 1000: REM EINDTITEL
270 GOSUB 5000: REM VERVAGEN
280 END

```

Het hele programma is nu gereed.

Indien u alle subroutines en de supervisor heeft ingetypt, kan het programma worden gerund.

Bij het begin van de meeste subroutines wordt het scherm schoongemaakt.

Aan het einde van de subroutines zien we een pauze-lus, deze dient voor het vasthouden van de tekst op het scherm.

Het programma kan nog worden uitgebreid door hier of daar muziek in te brengen.

Taak

16a

Laad het programma 'Briefkaart' van de oefencassette en RUN het programma.

Zet de computer niet af als u klaar bent met het programma. Na dit hoofdstuk wordt het namelijk tijd de **listing** van de programma's op de cassettebandjes te gaan bekijken. Druk na het einde van het programma op de RESET-toets en type:

```
LIST
```

U ziet dan een lange lijst regels voorbij schieten en merkt dan dat het programma te lang is om in een keer te worden gelIST.
We zullen het programma in delen splitsen. Type:

```
LIST 100,600
```

Het valt op dat de regels 100 tot en met 600 in hoofdzaak REM-instructies zijn die dienen om het programma duidelijk te maken.
Regel 220 opent het toetsenbord voor een latere GET-instructie.
Regel 230 geeft de nodige DIM-instructies voor de strings in het programma.
Regel 660 laat het programma eindigen.

Type nu:

```
LIST 1000,1270
```

Dit is een gedeelte van de titel van het programma.
Regel 1110 en 1120 zorgen voor veranderingen van kleuren.
Regel 1130 laat de cursor verdwijnen.

Type:

```
LIST 1270,1350
```

Het vervolg van de titel.
Regel 1350 RETURN stuurt het programma terug naar 610.

Type:

```
LIST 2000,2200
```

In deze subroutine zijn de gegevens van de briefkaart opgenomen.
Regel 2200 RETURN zorgt weer voor de terugkeer naar de supervisorregel 620.

Type:

```
LIST 3000,3170
```

Dit programmadeel opent de mogelijkheid gegevens in te voeren.
Regel 3170 RETURN stuurt terug naar regel 630.

Type:

```
LIST 4000,4260
```

Bekijk dit gedeelte en type daarna:

```
LIST 4270,4400
```

Deze beide delen vormen de eigenlijke briefkaart.
Regel 4400 RETURN zorgt weer voor het vervolg via regel 640 van de supervisor.

LIST 5000,5240

Dit programmadeel is de keuzemogelijkheid: opnieuw bekijken – nieuwe gegevens invoeren – opnieuw starten of eindigen.

Regel 5200 is de GET-instructie die de ASCII-code van de toetsen 1, 2, 3 en 4 bekijkt en aan de hand van het gevonden getal het programma laat vervolgen met de regels 5210, 5220, 5230 of 5240.

Regel 5240 RETURN brengt het programma weer bij de supervisorregel 650.

Type:

LIST 6000,7000

Het laatste gedeelte. Hierdoor wordt het programma beëindigd.

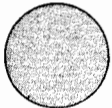
Nadat de kleuren veranderd zijn in de regels 6120, 6130 en 6140 wordt via 6160 RETURN het programma naar 660 END gestuurd.

In het programma bent u ook de instructie GRAPHICS 2 tegengekomen. Dit is een van de tekenmogelijkheden van de ATARI die in de volgende hoofdstukken zullen worden behandeld.

Het is raadzaam ook de overige programma's van de oefencassette te LISTen en te bekijken.

Er staan waardevolle ideeën in voor uw eigen programma's.

Taak 16a afgerond



GRAPHICS 0, 1 en 2

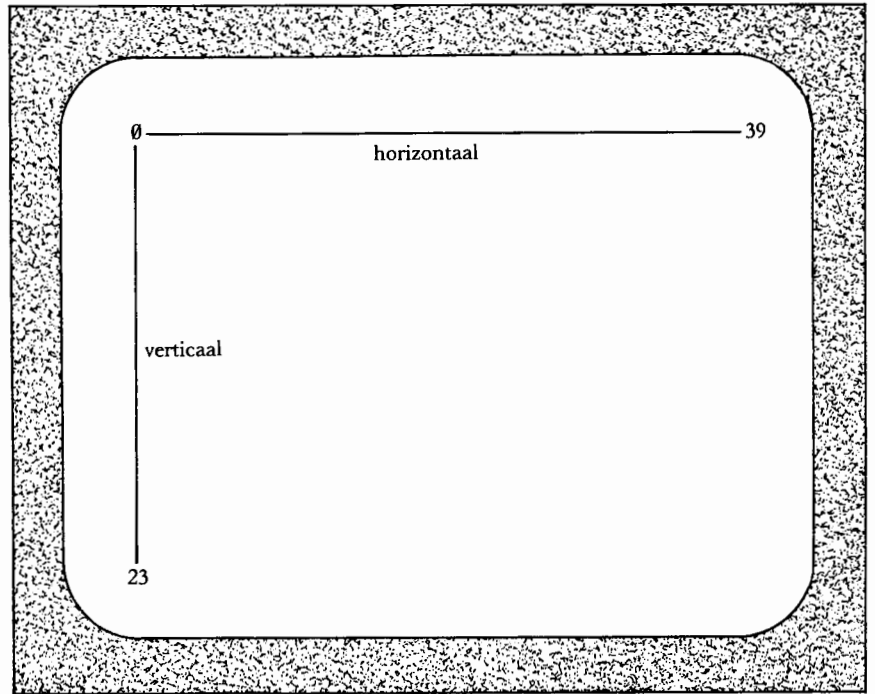
In de laatste hoofdstukken van dit boek worden twaalf verschillende grafische modes behandeld, namelijk GRAPHICS 0 tot en met 11.

17.1 GRAPHICS 0

Met de grafische mode GRAPHICS 0 (afgekort GR. 0) hebben we al uitgebreid kennis gemaakt.

Dit is namelijk de mode waarin we tot nu toe steeds hebben gewerkt. We zullen even de voornaamste punten van GRAPHICS 0 herhalen.

- 1 In de GRAPHICS 0-mode kunnen 40 tekens horizontaal (kolommen) en 24 tekens verticaal (rijen) worden geplaatst.



- 2 In de SETCOLOR-instructie kunnen de volgende kleuregisters worden gebruikt:
1 voor de tekens – 2 voor de achtergrond – 3 voor het kader.
Tevens kunnen alle kleurinstellingen van 0 tot en met 15 en alle kleurhelderheidsinstellingen van 0 tot en met 14 in de instructie worden geplaatst.
- 3 Alle tekens, letters, cijfers en grafische symbolen kunnen worden gebruikt.

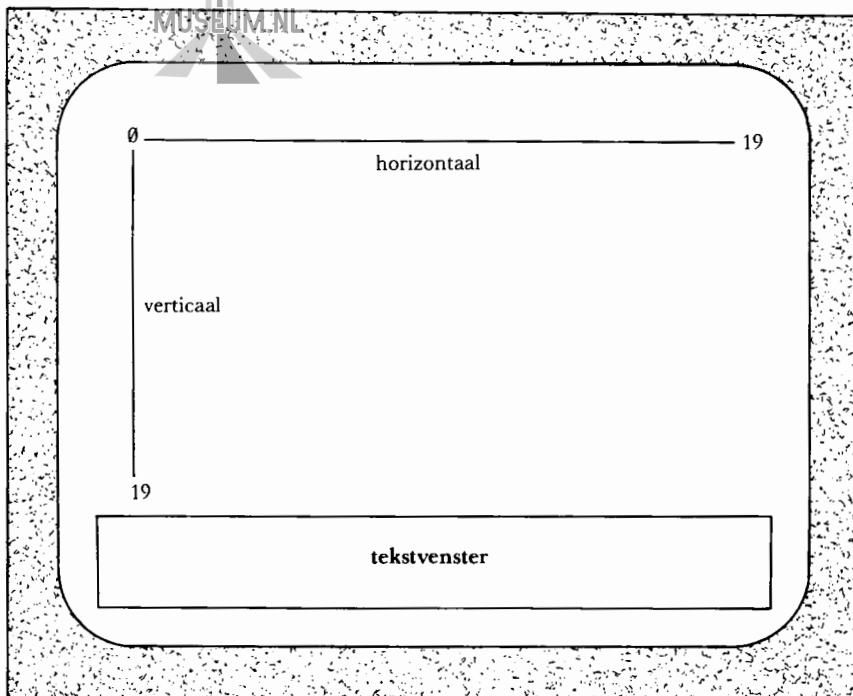
De computer wordt in GRAPHICS 0-mode gezet door:

- 1 Het normaal aanzetten van de computer.
- 2 Het indrukken van de RESET-toets indien in andere grafische modes werd gewerkt.
- 3 Het intypen van GRAPHICS 0. Deze methode wordt gebruikt indien in een programma in een andere grafische mode over moet worden gesprongen naar het normale programma.

17.2 GRAPHICS 1

De grafische mode 1 wijkt wat het aantal kolommen betreft af van GRAPHICS 0.

In GRAPHICS 1 hebben we twee verschillende beeldscherminstellingen. De eerste instelling krijgen we door GRAPHICS 1 te typen en op de RETURN-toets te drukken. Het scherm ziet er dan als volgt uit:



Het aantal tekens dat in de kolommen (horizontaal) kan worden geplaatst, is 20 (0 tot en met 19). Ook in de rijen (verticaal) kunnen twintig tekens worden geplaatst.

Het onderste gedeelte van het scherm is het zogenaamde tekstvenster. Het kader om het bovengedeelte heeft dezelfde kleur als het bovendeel.

Indien nu iets wordt getypt komt dat in het tekstvenster te staan. Daar dit stukje scherm maar vier regels hoog is, zal de tekst vrij snel oprollen. Op het bovendeel van het scherm wordt de tekst geprint in een letter die tweemaal zo hoog en even breed is als een normale letter.

Het intypen van een programma in GRAPHICS 1 gaat het makkelijkst op een normaal scherm. Dit krijgt u door de RESET-toets in te drukken. De instructie GRAPHICS 1 kan dan in een regel van het programma worden opgenomen.

Voor het printen op het bovenste deel van het scherm is een speciale PRINT-instructie nodig. Type en RUN:

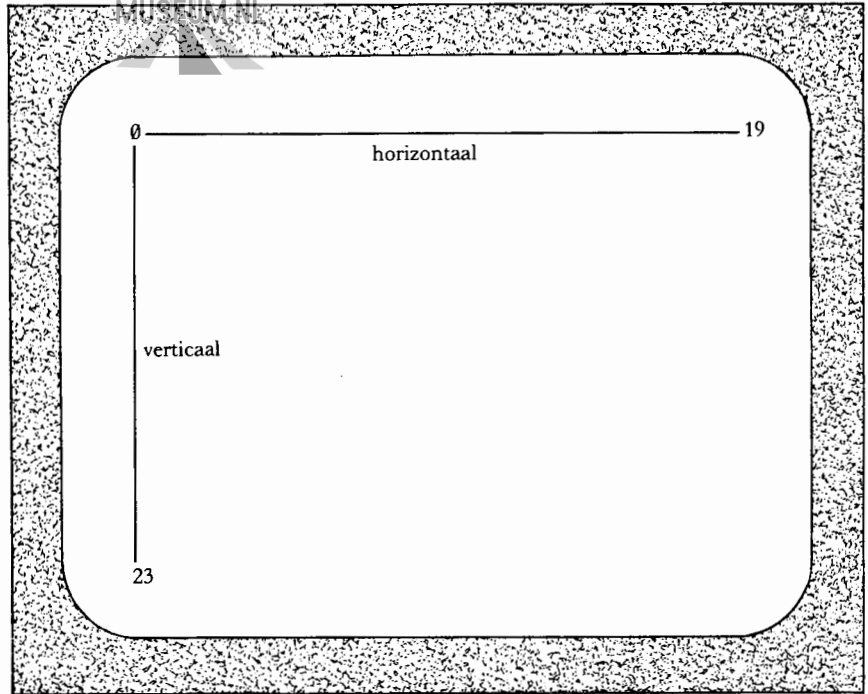
```
10 GR. 1
20 PRINT#6;" GROTE LETTERS"
30 PRINT"DIT STAAT IN HET TEKSTVENSTER"
```

U ziet dat alleen de tekst achter PRINT#6; op het bovenste deel van het scherm komt, de rest van het programma staat in het tekstvenster.

Indien u alleen met grote letters wilt werken, kunt u het tekstvenster laten verdwijnen door de tweede instelmogelijkheid van GRAPHICS 1 te gebruiken. Type dan:

```
10 GR. 1+16
```

GR. 1+16 is hetzelfde als GR. 17. Om verwarring met andere grafische modes te voorkomen zullen we echter in dit boek GR. 1+16 gebruiken. Het scherm heeft dan het onderstaande beeld:



U ziet dat er verticaal vier rijen bij zijn gekomen.

Bij het gebruik van GRAPHICS 1 + 16 moet rekening worden gehouden met het feit dat het scherm wordt gewist zodra het programma is afgelopen. De meeste tekst verdwijnt zo snel dat er niets te zien is dan een flits.

Dit kan worden ondervangen door als laatste regel een pauze-lus in het programma op te nemen:

```
25 FOR P=1 TO 1000: NEXT P
```

of een GOTO-instructie naar het eigen regelnummer:

```
25 GOTO 25
```

De pauze moet aangepast zijn aan de tijd die het programma op het scherm moet blijven staan. De GOTO-regel wordt op de normale manier opgeheven.

17.3 Kleuren in GRAPHICS 1

De GRAPHICS 1-mode heeft standaardkleuren: de achtergrondkleur is zwart en de kleur van de letters is oranje. De letterkleur kan worden veranderd zonder een SETCOLOR-instructie te gebruiken. Het enige dat hoeft te worden gedaan is in de PRINT-instructie hoofdletters of kleine letters, normaal of in reverse video, te gebruiken.

Type het onderstaande programma en bekijk het resultaat (de letters tussen de aanhalingstekens typen zoals in de REM-instructie is aangegeven). Type en RUN:

```
10 GR. 1+16
20 ?#6;"ORANJE": REM HOOFDLETTERS
30 ?#6;"licht groen": REM KLEINE LETTERS
40 ?#6;"█DONKERBLAUW█": REM HOOFDLETTER
S IN REVERSE VIDEO
50 ?#6;"█rood█": REM KLEINE LETTERS IN
```

Buiten deze standaardkleuren kan ook de SETCOLOR-instructie worden gebruikt in de GRAPHICS 1-mode. De kleurregisternummers wijken af van de kleurregisters in de GRAPHICS 0-mode.

- Register 0 om de tekstkleur te wijzigen die in hoofdletters wordt getypt.
- Register 1 om de tekstkleur te wijzigen die in kleine letters wordt getypt. Bepaalt tevens de helderheid van de tekst in het tekstvenster.
- Register 2 om de tekstkleur te wijzigen van hoofdletters die in reverse video worden getypt. Bepaalt tevens de kleur van het tekstvenster.
- Register 3 om de tekstkleur te wijzigen van de kleine letters die in reverse video worden getypt.
- Register 4 om de achtergrondkleur te wijzigen.

In de praktijk zult u voor vergrote tekst in hoofdzaak register 0 en 4 gebruiken.

De POKE-instructie kan worden gebruikt in de GRAPHICS 1-mode.

```
SETCOLOR 0,-,- = POKE 708,--
SETCOLOR 1,-,- = POKE 709,--
SETCOLOR 2,-,- = POKE 710,--
SETCOLOR 3,-,- = POKE 711,--
SETCOLOR 4,-,- = POKE 712,--
```

Achter de POKE-instructie kunnen de getallen worden geplaatst die in Aanhangsel E zijn opgenomen.

In onderstaand programma ziet u verschillende achtergrondkleuren. Type en RUN:

```
10 GR. 1+16: SE. 0, 0, 2
20 ?#6; "KLEUR ACHTERGROND"
30 FOR A=0 TO 15
40 SE. 4, A, 8
50 FOR P=1 TO 150: NEXT P
60 NEXT A
```

De regels 30, 40 en 60 plaatsen alle kleuren in de SETCOLOR-instructie voor de achtergrond.

Het volgende programma verandert de letterkleur. Type en RUN:

```
10 GR. 1+16
20 ?#6; "KLEUR LETTERS"
30 FOR A=0 TO 15
40 SE. 0, A, 10
50 FOR P=1 TO 150: NEXT P
60 NEXT A
```

De regels 30 tot en met 60 plaatsen de kleuren in de SETCOLOR-instructie voor de letters.

In GRAPHICS 1-mode kan gebruik worden gemaakt van de normale instructies en commando's. Bij de POSITION-instructie moet wel rekening worden gehouden met het aantal kolommen, dat maar de helft van de

GRAPHICS 0-mode is. Een van de meest voorkomende foutmeldingen bij de POS.-instructie is dan ook ERROR 141. Deze foutmelding geeft aan dat u buiten het bereik van de grafische mode komt.

GRAPHICS 1 kan in de mode voor grafische symbolen en kleine letters worden gezet door te typen:

```
POKE 756,226
```

Voor terugkeer naar de normale tekens typt u:

```
POKE 756,224
```

Type en RUN het onderstaande programma (de grafische symbolen krijgt u op de gebruikelijke wijze: de CONTROL-toets ingedrukt houden en de gewenste toets indrukken).

```
1 Ø GR. 1
2 Ø POKE 756,226
3 Ø ?#6;"|H|J HOOFDLETTERS |H|J"
```

U ziet nu op het scherm het woord 'hoofdletters' in kleine letters staan met aan weerszijden de grafische symbolen. De achtergrond is opgebouwd uit hartjes.

Indien u de hartjes en de tekst kwijt wilt, type dan:

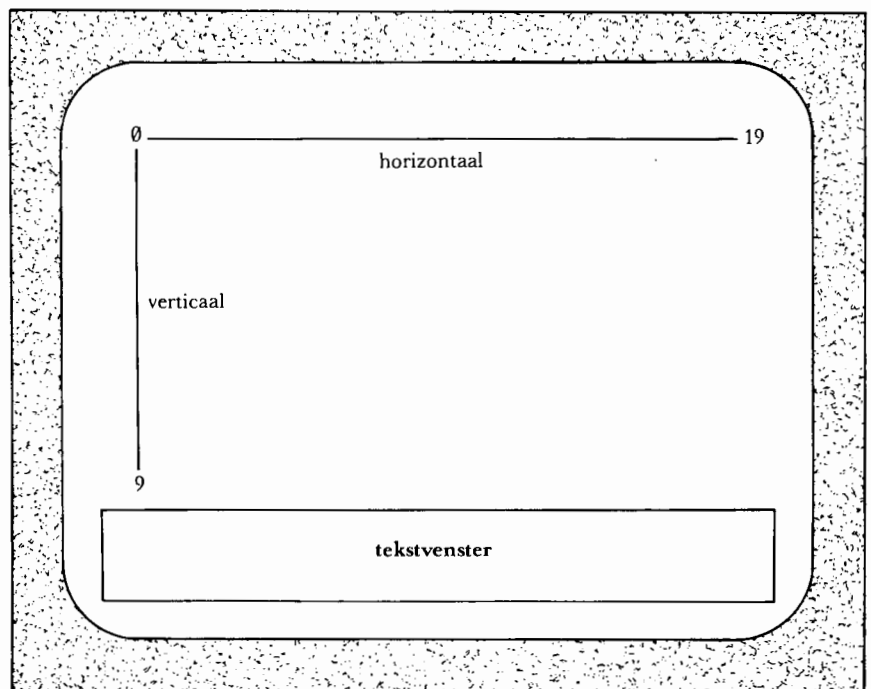
```
25 SETCOLOR Ø,Ø,Ø
```

De achtergrond, hartjes en letters krijgen nu alle dezelfde kleur. Alleen de grafische symbolen blijven over.

17.4 GRAPHICS 2

De derde grafische mode is GRAPHICS 2. Het verschil tussen GRAPHICS 1 en 2 is alleen de grootte van de tekens.

Het scherm in GRAPHICS 2-mode is ingedeeld zoals hieronder is aangegeven.



De tekens zijn in de GRAPHICS 2-mode tweemaal zo hoog als in de GRAPHICS 1-mode, de breedte is echter gelijk.

Het verwijderen van het tekstvenster geschiedt op dezelfde wijze als bij GR. 1. Type:

```
10 GR. 2+16
```

De grafische modes 1 en 2 worden dikwijls gebruikt voor een titel van een programma.

Een manier om een titel zonder veel rekenwerk midden op het scherm te plaatsen, is het gebruik van de LEN-functie.

Als u het woord 'MIDDEN' midden op het scherm wilt plaatsen, type dan:

```
10 DIM A$(8)
20 A$="MIDDEN"
30 GR. 2      (of GR. 1)
40 POS. (20-LEN(A$))/2,4
50 ?#6;A$
```

Hieronder volgt een programma waarin de GR. 0-, GR. 1- en GR. 2-modes worden gebruikt. Type en RUN:

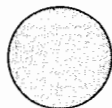
```
10 ?"\" : DIM NAAM$(20)
20 ?"TYPE UW NAAM EN DRUK OP RETURN"
30 POS. 5,6: ?"NAAM";:INPUT NAAM$
40 GR. 1+16
50 POS. (20-LEN(NAAM$))/2,10
60 ?#6;NAAM$
70 FOR P=1 TO 1000: NEXT P
80 GR. 2+16
90 POS. 7,5
100 ?#6;"EINDE"
110 FOR A=0 TO 14
120 SE. 0,4,A
130 SE. 4,4,14
140 FOR P=1 TO 200: NEXT P
150 NEXT A
160 GR. 0
```

Taak

17a

Laad het programma 'Welke is het' van de oefencassette.
Dit programma is een quiz over de hoofdstukken 14, 15, 16, en 17.

Taak 17a afgerond



GRAPHICS 3, 5 en 7

De grafische modes die in dit hoofdstuk worden behandeld zijn echte tekenmodes. U kunt alleen tekst opnemen in het tekstvenster, tenzij het een getekende tekst is.

18.1 Kleuren

Alvorens aan de grafische modes te beginnen zullen we eerst de kleurmogelijkheden behandelen in deze drie grafische modes.

De GRAPHICS 3, 5 en 7 kennen ook weer een stel standaardkleuren die in vier kleurregisters zijn ondergebracht. Deze kleurregisters zijn:

0	oranje	
1	lichtgroen	bepaalt tevens de helderheid van de tekens in het tekstvenster
2	donkerblauw	bepaalt tevens de kleur van het tekstvenster
4	zwart	de standaardkleur voor de achtergrond

De kleuren uit deze registers kunnen op de normale manier worden veranderd door de SETCOLOR-instructie: SETCOLOR 0,7,4 verandert de kleur uit register 0 in blauw; SETCOLOR 1,0,14 verandert de kleur uit register 1 in wit, enzovoort.

Tevens kunnen de registers ook weer door de POKE-instructie worden veranderd:

```

SETCOLOR 0,-,- = POKE 708,--
SETCOLOR 1,-,- = POKE 709,--
SETCOLOR 2,-,- = POKE 710,--
SETCOLOR 4,-,- = POKE 712,--

```

Achter de POKE-instructie kunnen de getallen worden geplaatst die zijn opgenomen in Aanhangsel E.

18.2 De COLOR-instructie

Door middel van de SETCOLOR- of de POKE-instructie wordt de kleur in het kleurregister geplaatst van de grafische modes die in dit hoofdstuk worden behandeld. De gekozen kleuren worden echter in het GRAPHICS-programma gebracht door de COLOR-instructie (afgekort C.). De COLOR-instructie bepaalt welk register wordt gebruikt. De enige moeilijkheid is dat het nummer achter de COLOR-instructie niet correspondeert met de kleurregisternummers, zoals hieronder te zien is:

```

COLOR 0 is kleurregister 4,
COLOR 1 is kleurregister 0,
COLOR 2 is kleurregister 1 en
COLOR 3 is kleurregister 2.

```

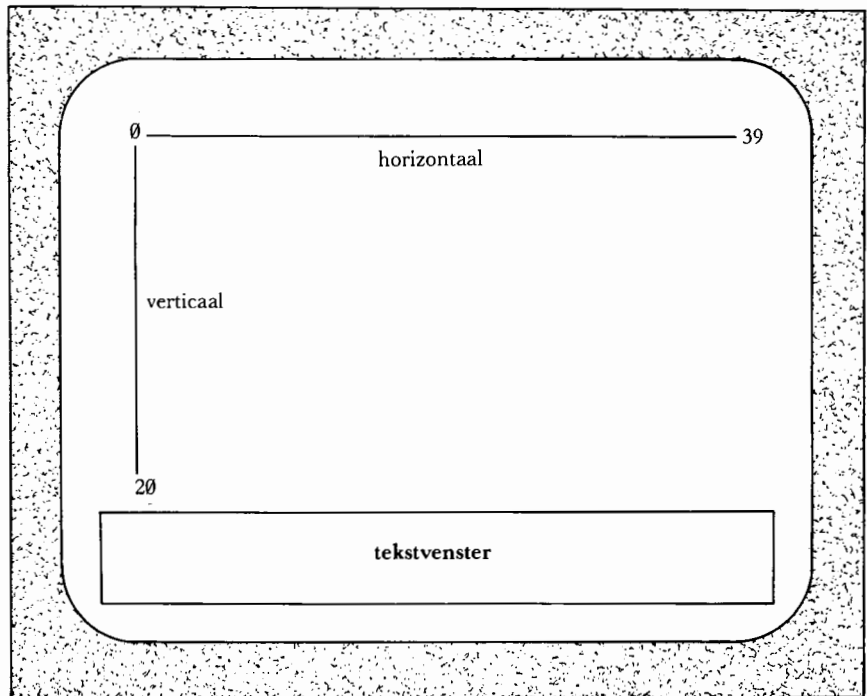
Samengevat: de kleur in het register wordt bepaald door:

- a de standaardkleur die ingebracht wordt bij het aanzetten van de computer;
- b de SETCOLOR- of de POKE-instructie.

De kleur wordt in het GRAPHICS-programma gebracht door de COLOR-instructie.

Het lijkt moeilijker dan het is, maar bij het bespreken van de programma's wordt het een stuk duidelijker.

GRAPHICS 3 heeft dezelfde beeldverdeling als GRAPHICS 0: 40 tekens horizontaal (kolommen) en 20 tekens verticaal (rijen) met tekstvenster en 24 tekens verticaal zonder tekstvenster.



GRAPHICS 3 bezit geen kader. Het is een echte tekenmode, want er kan alleen met gekleurde blokjes ter grootte van de cursor worden gewerkt. Het eerste dat moet worden gedaan is de plaats bepalen (plotten) waar het eerste blokje (de cursor) moet komen te staan. Dit wordt gedaan door de PLOT-instructie toe te passen (PLOT kan worden afgekort tot PL.). De PLOT-instructie heeft de volgende vorm:

PLOT X, Y

waarbij X het getal is dat de kolom aangeeft en Y het getal dat de rij aangeeft.

Stel dat u het eerste blokje in de 4e kolom op de 3e rij wilt plaatsen, dan luidt de instructie als volgt:

PLOT 4, 3

Het tweede punt dat bepaald moet worden is de plaats waarheen een lijn (bestaande uit aaneengeschaalde blokjes) moet worden getrokken. Dit wordt gedaan door de DRAWTO(teken-naar)-instructie (DRAWTO wordt afgekort tot DR.).

De DRAWTO-instructie heeft de volgende vorm:

DRAWTO X, Y

waarbij X en Y weer de getallen zijn die de kolom en rij aangeven.

Stel dat we de lijn, waarvan het beginpunt door de instructie PLOT 4,3 is bepaald, horizontaal willen trekken naar kolom 30, dan wordt de DRAWTO-instructie:

DRAWTO 30, 3



In dit programma komt dan te staan:

```
30 PLOT 4,3
40 DRAWTO 30,3
```

Daarvoor moet dan de GRAPHICS 3-instructie worden geplaatst:

```
10 GRAPHICS 3
```

en, om de kleur te bepalen, de COLOR-instructie. Bijvoorbeeld een oranje lijn:

```
20 COLOR 1
```

Type en RUN nu het hele programma:

```
10 GR. 3
20 COLOR 1
30 PLOT 4,3
40 DRAWTO 30,3
```

Als u dit onder de knie heeft, geeft het tekenen verder geen moeilijkheden meer.

Wilt u vanuit het eindpunt van de lijn een lijn recht naar beneden trekken naar de 15e rij, dan wordt de volgende regel:

```
50 DRAWTO 30,15
```

waarbij 30 de kolom aangeeft en 15 de rij, waarin het eindpunt van de tweede lijn moet komen.

Moet deze lijn een andere kleur hebben? Geen bezwaar, maak de regel:

```
45 COLOR 2
```

Nu gaan we van de twee lijnen een rechthoek maken. De onderste lijn loopt dan naar de 4e kolom op de 15e rij, dus:

```
60 DRAWTO 4,15
```

De laatste verticale lijn loopt naar het beginpunt:

```
70 DRAWTO 4,3
```

Laten we nu de kleurverandering in regel 45 eruit, dan wordt het programma:

```
10 GR. 3
20 COLOR 1
30 PLOT 4,3
40 DRAWTO 30,3
50 DRAWTO 30,15
60 DRAWTO 4,15
70 DRAWTO 4,3
```

Type en RUN dit programma en bekijk het resultaat.

Op één regel kunnen natuurlijk ook meerdere instructies, gescheiden door de dubbele punt, worden opgenomen. 139

Bijvoorbeeld:

```
30 PL. 4,3: DR. 30,3: DR. 30,15: DR. 4,
15: DR. 4,3
```

waarmee de hele tekening in één regel is ondergebracht.

Ook diagonaal kunnen lijnen worden getrokken, zoals in onderstaand programma, waarin we tevens het tekstvenster laten verdwijnen. Daarin moet ook weer een pauze- of GOTO-regel aan het programma worden toegevoegd om het beeld vast te houden.

Type en RUN:

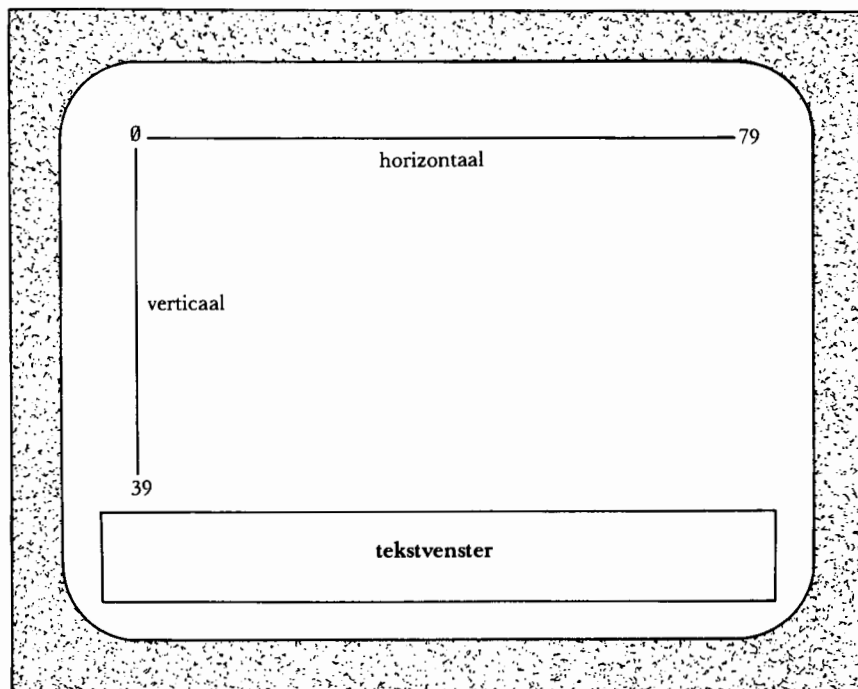
```
10 GR. 3+16
20 COLOR 3
30 PL. 7,0: DR. 31,23
40 FOR P=1 TO 1500: NEXT P
```

U ziet dat deze lijn echt uit blokjes is opgebouwd. Scherpe lijnen komen we later tegen in de hogere grafische modes.

Voor het tekenen in de grafische modes is de makkelijkste manier eerst een tekening te maken op ruitjespapier, in de verhoudingen van het scherm (bij GRAPHICS 3 dus op 40×24 ruitjes). U kunt dan gemakkelijk de PLOT- en DRAW-punten bepalen.

18.4 GRAPHICS 5

GRAPHICS 5 wijkt alleen wat de schermindeling betreft af van GRAPHICS 3: u heeft 80 horizontale kolommen en 40 verticale rijen mét tekstvenster óf 48 verticale rijen zónder tekstvenster.



De tekens zijn dus half zo hoog en half zo breed als die van GRAPHICS 3.

GRAPHICS 5 is daarom geschikt voor meer gedetailleerde tekeningen. In deze mode gelden ook alle kleurmogelijkheden en instructies die voor GRAPHICS 3 van toepassing zijn.

Hieronder een programma dat u het verschil laat zien met de GRAPHICS 3-mode. Type en RUN:

```

10 GR . 5+16
20 COLOR 3
30 PLOT 3,5: DR. 70,5: DR. 3,45: DR. 3,5
40 GOTO 40

```

Een paar programmaregels die ook goed te gebruiken zijn in een GRAPHICS-programma zijn de volgende:

```

C=INT(RND(0)*4)
COLOR C

```

Deze regels wijzigen geheel willekeurig – door de RND-functie – bij iedere passage van de lus het nummer achter de COLOR-instructie. Een voorbeeld van deze regels in een programma is:

```

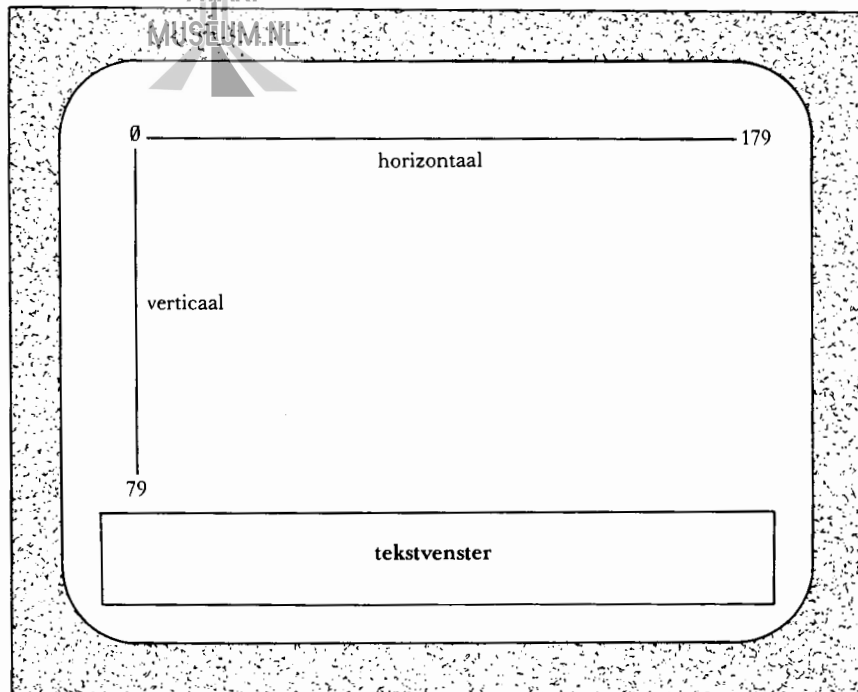
10 GR . 5
20 C=INT(RND(0)*4)
30 COLOR C
40 X=INT(RND(0)*79)
50 Y=INT(RND(0)*39)
60 PL . X,Y
70 GOTO 20

```

Als u dit programma intypt en RUNt, wordt het hele scherm gevuld met verschillend gekleurde blokjes. Deze blokjes worden op willekeurige plaatsen op het scherm gezet, doordat in de PLOT-instructie X en Y ook beide door de RND-functie worden bepaald.

18.5 GRAPHICS 7

Ook bij deze grafische mode is alleen de schermindeling anders dan bij GRAPHICS 3 en 5: u beschikt over 160 horizontale kolommen en 80 verticale rijen met tekstvenster of 96 verticale rijen zonder tekstvenster.



Met deze mode kan dus weer gedetailleerder worden getekend dan met GR. 5.

Voor het overige zijn voor GRAPHICS 7 ook weer alle kleurmogelijkheden en instructies van toepassing die voor GRAPHICS 3 gelden.

Hieronder volgt een programma dat u de verschillen tussen de grafische modes laat zien, in een en dezelfde tekening uitgevoerd in GR. 7; GR. 5 en GR. 3. Deze tekening is het handelsmerk van ATARI.

Type en RUN:

```

10 GR. 7: COLOR 1
20 PL. 35,1: DR. 40,1: DR. 35,8
30 GOSUB 200
40 GR. 5: COLOR 1
50 PL. 40,1: DR. 35,1: DR. 35,3: DR.40,
3: DR. 40,7: DR. 35,7
60 GOSUB 200
70 GR. 3: COLOR 1
80 PL. 35,1: DR. 39,1: DR. 39,7: DR. 35
,7: PL. 35,4: DR. 39,4
90 GOSUB 200
100 END
200 PL. 17,5: DR. 17,9: DR. 9,17: PL. 19
,5: DR. 19,17
210 PL. 20,5: DR. 20,17: PL. 22,5: DR. 2
2,9: DR. 30,17
220 FOR P=1 TO 1500: NEXT P
230 RETURN

```

Probeer, voor u het programma RUNt, uit te zoeken wat de tekening in regel 200 en 210 voorstelt.

Hierna volgt een programma voor een onweersbui, waarin grafische-, licht- en geluidseffecten zijn opgenomen.

Type en RUN:

```

10 GR. 7+16
20 SE. 4,0,0: SE. 1,0,4: COLOR 2
30 PL. 0,0: DR. 15,25: DR. 60,40: DR.
50,60: DR. 90,70: DR. 110,90
40 FOR TH=0 TO 255: SOUND 0,TH,8,12
50 IF TH=15 THEN SE. 1,0,14
60 NEXT TH: SE. 1,0,0
70 FOR P=1 TO 200: NEXT P
80 GOTO 10
  
```

In regel 20 wordt de achtergrond en de kleur van de lijn ingesteld.

Regel 30 is de tekening van de bliksemflits (wilt u een andere flits, wijzig dan deze getallen).

Regel 40 brengt alle 256 toonhoogten in de SOUND-instructie.

Regel 50 zorgt dat de flits helderder wordt als TH 15 is.

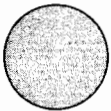
Taak

18a

Laad het programma 'Tekenen' van de oefencassette.

In dit programma kunt u de computer verschillende figuren laten tekenen. Bij het tekenen van een cirkel (die in hoofdstuk 19 wordt behandeld) legt de computer een aantal keren dezelfde omtrek af. Het tekenen van de cirkel is voltooid als in het midden van de cirkel een stip verschijnt. Daarna kunt u door het indrukken van de START-toets weer terugkeren naar het menu.

Taak 18a afgerond



GRAPHICS 4, 6 en 8



De grafische modes 5 en 7 gebruiken vrij veel geheugenruimte omdat ze over verschillende kleurmogelijkheden beschikken. Indien – gezien de lengte van het programma – niet veel geheugen over is, of omdat er toch slechts in één kleur wordt gewerkt, is het verstandiger een andere grafische mode te kiezen. De GRAPHICS 4 en 6 hebben veel minder geheugenruimte nodig dan de vergelijkbare GRAPHICS 5 en 7.

19.1 GRAPHICS 4

GRAPHICS 4 is wat de schermindeling en alle andere mogelijkheden betreft geheel gelijk aan de GRAPHICS 5. De enige verschillen zijn de kleuren.

De standaardkleuren van GRAPHICS 4 zijn in vier kleurregisters ondergebracht. Deze registers zijn:

0	oranje	kleur tekeningen
1	kleur gelijk aan kleur tekstvenster, maar met grotere helderheid	helderheid van tekens in het tekstvenster
2	blauw	kleur tekstvenster
3	zwart	kleur achtergrond

Deze kleuren kunnen op de normale manier worden veranderd door de SETCOLOR-instructie, bijvoorbeeld SETCOLOR 0,7,8 verandert de tekenkleur uit register 0 in blauw en SETCOLOR 2,1,4 verandert de kleur van het tekstvenster in goud.

De kleuren in de registers kunnen ook worden veranderd door de POKE-instructie te gebruiken, met dien verstande dat POKE 709,-- alleen belangrijk is voor de helderheid van de tekeningen.

```
SETCOLOR 0,-,- = POKE 708,--
SETCOLOR 1,-,- = POKE 709,--
SETCOLOR 2,-,- = POKE 710,--
SETCOLOR 4,-,- = POKE 712,--
```

Er is echter slechts één mogelijkheid voor de COLOR-instructie: COLOR 1, en dit is kleurregister 0.

Alle tekeningen uit een programma in GRAPHICS 4 zijn dus altijd van dezelfde kleur. Het is wel mogelijk de kleur te veranderen, maar dan verandert alles dat op het scherm getekend staat in die kleur.

```
10 GR . 4+16
20 COLOR 1
30 PL . 20,15: DR . 60,15: FOR P=1 TO 500
: NEXT P
40 SE . 0,7,8
50 COLOR 1
60 PL . 20,25: DR . 60,25
70 GOTO 70
```

Regel 20 brengt de (standaard) oranje kleur in de lijn die in regel 30 wordt getekend.

Regel 40 verandert de standaardkleur in blauw; deze kleur wordt in regel 50 in het programma geplaatst. Zowel de lijn uit regel 60 als die uit regel 30 worden nu blauw.

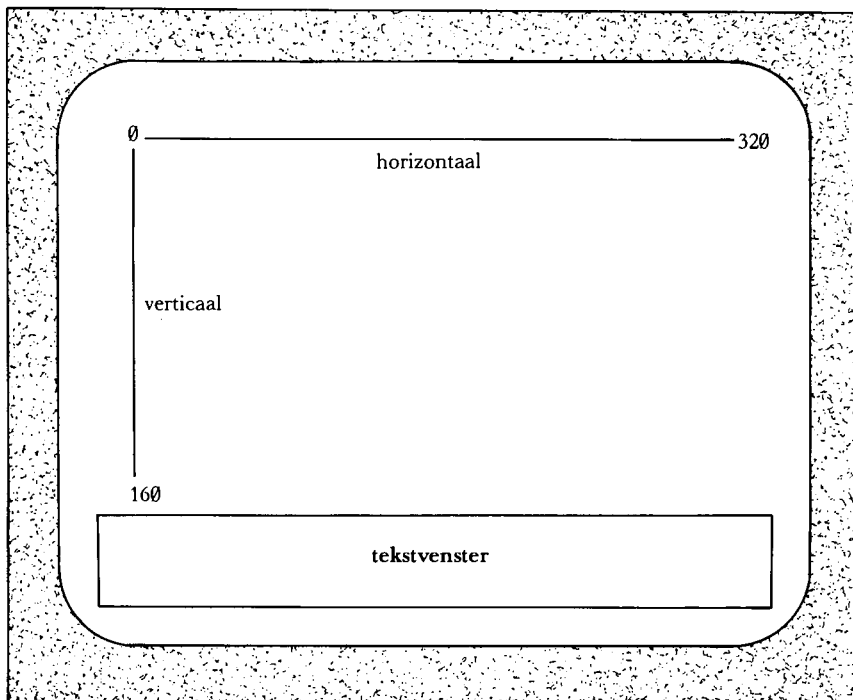
19.2 GRAPHICS 6

GRAPHICS 6 is wat schermindeling en alle andere mogelijkheden betreft, op de kleuren na, gelijk aan GRAPHICS 7. De kleurmogelijkheden zijn gelijk aan GRAPHICS 4.

19.3 GRAPHICS 8

GRAPHICS 8 is de ideale grafische mode voor gedetailleerd tekenwerk. De schermindeling is: 320 kolommen horizontaal en 160 rijen verticaal met tekstvenster of 192 rijen zonder tekstvenster.

GRAPHICS 8 heeft ook weer een kader om de achtergrond.



Deze indeling opent ongekennde mogelijkheden voor tekenwerk. De standaardkleuren zijn in drie registers ondergebracht:

- | | | |
|---|--------------------------------------|-----------------------------------|
| 1 | kleur gelijk aan de achtergrondkleur | kleur tekeningen |
| 2 | blauw | kleur achtergrond en tekstvenster |
| 3 | zwart | kaderkleur |

Deze kleuren kunnen op de normale manier met de SETCOLOR-instructie worden gewijzigd:

SETCOLOR 1,-,- verandert alleen de helderheid van de tekeningen.

SETCOLOR 2,-,- bepaalt de kleur van achtergrond, tekstvenster en tekeningen.

SETCOLOR 4,-,- bepaalt de kaderkleur.

Ook de POKE-instructie kan hierbij weer worden toegepast, met dien verstande dat bij POKE 709,-,- alleen de helderheid van de tekeningen verandert.

SETCOLOR 1,-,- = POKE 709,-,-

SETCOLOR 2,-,- = POKE 710,-,-

SETCOLOR 4,-,- = POKE 712,-,-

De COLOR-instructie kan alleen worden gebruikt om de helderheid van de tekening te bepalen. COLOR 1 is kleurregister 1. (Eindelijk een COLOR en kleurregister die een zelfde nummer hebben.)

Om een idee van de tekenmogelijkheden van GRAPHICS 8 te krijgen, volgt hier een programma voor een vijfpuntige ster. Type en RUN:

```

10 GR . 8+16
20 COLOR 1
30 PL . 160,10: DR . 110,140: DR . 230,60:
DR . 90,60: DR . 210,140: DR . 160,10
40 GOTO 40

```

De ster wordt gevormd in regel 30 en begint midden boven op de 10e rij. Probeer eerst de tekening op papier te maken.

Deze grafische mode is ook zeer geschikt om ronde of enigszins ronde voorwerpen te tekenen. Om dit te illustreren zullen we nu een programma maken voor het tekenen van een cirkel.

Voor een cirkel moeten normaal enige honderden plotpunten worden gemaakt, maar we kunnen ook gebruik maken van de eigenschappen van twee functies, de SIN- en de COS-functie.

Voor de opzet van dit boek is het niet belangrijk dat u weet wat de sinus en cosinus van hoeken is, u hoeft alleen maar te weten hoe u ze hier moet gebruiken.

Type en RUN:

```

10 GR . 8
20 SE . 2,0,0
30 COLOR 1
40 FOR L=1 TO 400
50 A=A+0.05
60 X=SIN(A)*50: Y=COS(A)*50
70 PLOT X+80,Y+55
80 NEXT L

```

We zullen dit programma regel voor regel bespreken.

10 stelt GRAPHICS 8-mode in.

20 maakt de achtergrondkleur zwart.

30 stelt de tekenkleur in.

40 maakt een lus om alle cirkelpunten te tekenen. Tekent u een grotere cirkel dan zijn er meer punten nodig, bij een kleinere minder. Het getal 400 moet dan overeenkomstig worden aangepast.

50 geeft de variabele A de eerste waarde. (A is de hoek waarvan de sinus en cosinus moeten worden bepaald.) Hoe kleiner de waarde van A is, hoe dichter de puntjes waaruit de cirkel wordt opgebouwd bij elkaar komen te liggen.

60 bepaalt de grootte van de PLOT-punten X en Y door de sinus- en cosinus-functies en het getal waarmee deze worden vermenigvuldigd (hier 50). Het getal geeft de afmeting van de cirkel. Voor een grotere cirkel moet u deze getallen groter maken en voor een kleinere kleiner.

Het kan zijn dat uw televisie iets afwijkt en dat daardoor de vertoonde cirkel niet precies rond is. Indien de cirkel verticaal afwijkt moet 50 achter $Y=\text{COS}(A)*$ kleiner worden, indien de cirkel horizontaal afwijkt moet 50 groter worden (deze verandering in kleine stapjes).

Wilt u een grotere of kleinere cirkel, dan beide getallen achter $X=(\text{SIN})A*$ en $Y=\text{COS}(A)*$ groter of kleiner maken. (Wel rekening houden met het aantal kolommen en rijen.) 147

70 bepaalt het middelpunt. Wilt u de cirkel naar links dan het getal 80 groter maken, naar rechts het getal 80 kleiner maken, omhoog het getal 55 vergroten en omlaag het getal 55 verkleinen.

We zullen nu de grootte van de cirkel veranderen. Type en RUN:

```
60 X=SIN(A)*30: Y=COS(A)*30
```

U kunt ook de teken tijd verkorten door regel 40 te wijzigen in:

```
40 FOR L=1 TO 200
```

Meerdere cirkels gelijktijdig is ook mogelijk. Type en RUN:

```
35 FOR B=1 TO 5
70 PLOT X+B*35, Y+55
85 NEXT B
```

We hebben gezien dat als de cirkel afgeplat is, dit wordt veroorzaakt doordat het getal achter Y te klein is. Deze wetenschap gebruiken we om een ellips te maken. Type en RUN:

```
10 GR. 8
20 COLOR 1
30 FOR L=1 TO 400
40 A=A+.05
50 X=SIN(A)*80: Y=COS(A)*40
60 PL. X+160, Y+80
70 NEXT L
```

Regel 50 zorgt voor de verandering.

Bij X wordt de sinus met 80 vermenigvuldigd, bij Y de cosinus slechts met 40.

De veranderingen in regel 60 bepalen gewoon een ander middelpunt voor de ellips.

Wijzig nu regel 60 in:

```
60 PL. X+160, Y+80: DR. 160, 80
```

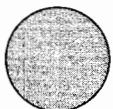
Deze regel zorgt ervoor dat de ellips wordt opgevuld, doordat uit het plot-punt van de ellips lijnen naar het middelpunt van de ellips worden getrokken.

Taak

19a

Laad het programma 'Sorteren' van de oefencassette. RUN dit programma en voer de opdrachten uit.

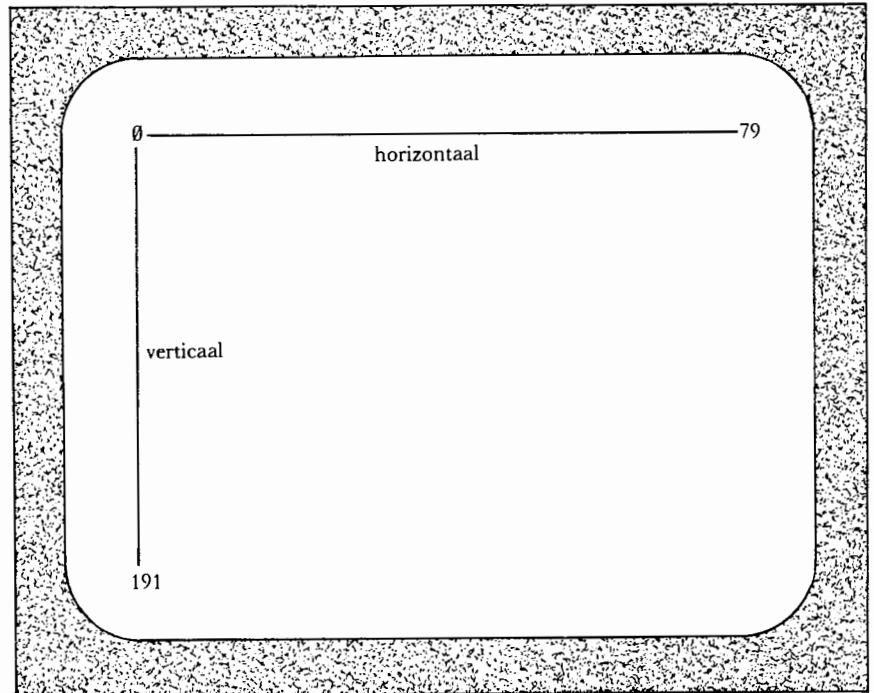
Taak 19a afgerond



GRAPHICS 9, 10 en 11

De ATARI 800X is uitgerust met de GTIA-chip, die het mogelijk maakt in deze drie grafische modes vele kleuren en tinten in die kleuren te gebruiken. In een van de drie modes heeft u zelfs alle 256 mogelijkheden om uit te kiezen.

De schermindeling voor alle drie grafische modes is gelijk: 80 kolommen horizontaal en 192 rijen verticaal. Dit is ongewoon, daar normaal het aantal beeldpunten horizontaal groter is dan verticaal. De beeldpunten hebben de vorm \square en zijn viermaal zo breed als hoog. De schermindeling is als volgt:



GRAPHICS 9, 10 en 11 bezitten geen tekstvenster, zodat dus niet in direct mode kan worden gewerkt. De laatste regel van ieder programma moet een pauze-lus of een GOTO naar de eigen regel bevatten.

De normale instructies en commando's die voor de grafische modes 4 tot en met 8 gelden, zijn ook van toepassing op deze drie modes.

20.1 GRAPHICS 9

De moeilijkheid bij deze mode is het selecteren van de juiste kleur. In GRAPHICS 9 is het mogelijk de 16 kleuren en in iedere kleur de 16 helderheidstinten gelijktijdig op het scherm te brengen. In totaal heeft u dus 256 mogelijkheden.

Als eerste moet de SETCOLOR-instructie worden gebruikt. GRAPHICS 9 heeft maar één kleurregister: 4. De SETCOLOR-instructie heeft de volgende vorm: SETCOLOR 4,X,0.

Op de plaats waar de letter X staat kunnen alle getallen van 0 tot en met 15 worden geplaatst. Deze getallen corresponderen met de kleurentabel in hoofdstuk 8. Het laatste cijfer, achter de komma na X, moet altijd 0 zijn.

De helderheidsfactor wordt ingebracht door de COLOR-instructie. Achter COLOR kan een getal van 0 tot en met 15 worden gezet. Hiervan is 0 de donkerste en 15 de lichtste tint.

Hieronder volgt een programma waarin 16 verticale blokken op het scherm worden gebracht in 5 verschillende helderheidsfactoren, die daarna alle 16 kleuren aannemen. Type en RUN:

```

10 GR. 9
20 FOR R=0 TO 15
30 FOR H=4 TO 12 STEP 2
40 COLOR H
50 PL. 5*R+H/2-2,0: DR. 5*R+H/2-2,191
60 NEXT H: NEXT R
70 FOR K=0 TO 15
80 SE. 4,K,0
90 FOR P=1 TO 750: NEXT P
100 NEXT K

```

In regel 50 ziet u dat in de PLOT- en DRAWTO-instructies ook berekeningen kunnen worden opgenomen. PLOT loopt hier van $(5 \times 0) + (4:2) - 2 = 0$ tot DRAWTO $(5 \times 15) + (12:2) - 2 = 79$ en dat zijn alle kolommen.

Hieronder volgt een programma dat door de RND-functie te gebruiken een willekeurige tekening op het scherm brengt, die alle kleuren en schakeringen doorloopt.

```

10 GR. 9
20 FOR A=0 TO 15: SE. 4,A,0
30 FOR B=0 TO 12: COLOR B
40 C=INT(RND(0)*A*B)
50 PL. 40,C: DR. 0,A*B: PL. 40,C: DR. 7
9,A*B
60 FOR P=1 TO 50: NEXT P
70 NEXT B: NEXT A
80 GOTO 80

```

Probeer zelf aan de hand van regel 50 uit te zoeken hoe deze lijn gaat lopen.

20.2 GRAPHICS 10

In GRAPHICS 10 kunt u uit acht kleuren en alle tinten van die kleuren kiezen. Tevens kan aan de achtergrond nog een extra kleur worden toegekend.

Het nadeel is dat GRAPHICS 10 de meest moeilijke grafische mode is om kleuren en tinten aan toe te voegen. Er is namelijk een COLOR-, POKE- en/of SETCOLOR-instructie voor nodig.

GRAPHICS 10 heeft 5 kleurregisters:

register 0	oranje	tekenkleur
register 1	groen	tekenkleur
register 2	blauw	tekenkleur
register 3	roze	tekenkleur
register 4	zwart	kaderkleur

Deze kleuren kunnen op de normale manier door de SETCOLOR-instructie worden gewijzigd. SETCOLOR 0,10,8 bijvoorbeeld verandert oranje in turquoise.

In GRAPHICS 10 kunnen acht verschillende COLOR-instructies worden gebruikt. In de eerste vier instructies kan alleen door een POKE-instructie een andere kleur worden ingebracht. COLOR 0 verandert de achtergrondkleur, COLOR 8 de kaderkleur; de overige zijn tekenkleuren. Hieronder volgt een tabel met de SETCOLOR- en de POKE-waarden.

Color	Setcolor	Poke
0	geen	704,--
1	geen	705,--
2	geen	706,--
3	geen	707,--
4	0,-,-	708,--
5	1,-,-	709,--
6	2,-,-	710,--
7	3,-,-	711,--
8	4,-,-	712,--

Hieronder een programma dat u enige mogelijkheden van de GRAPHICS 10 laat zien. Het loont de moeite zelf de verschillende COLOR-instructies uit te proberen, waarbij u door POKE- of SETCOLOR-instructies andere kleuren in de COLOR-instructies brengt.

```

10 GR. 10
20 FOR X=0 TO 8: COLOR X
30 FOR Y=0 TO 23: PL. 0,0: DR. 79,X*Y
40 NEXT Y: NEXT X
50 FOR A=704 TO 712
60 B=INT(RND(0)*255)
70 POKE A,B: NEXT A
80 GOTO 20

```

In regel 20 worden achter COLOR de getallen 0 tot en met 8 geplaatst. Regel 30 tekent lijnen van links boven naar rechts, afhankelijk van de waarden van X en Y.

In regel 60 wordt een willekeurig getal, tussen 0 en 255, door de RND-functie bepaald.

In regel 70 wordt dit getal in een POKE-instructie geplaatst.

De lus 70 - 50 doorloopt alle POKE-adressen van 704 tot en met 712.

Via regel 80 wordt de nieuwe kleur in regel 20 geplaatst.

20.3 GRAPHICS 11

GRAPHICS 11 biedt de mogelijkheid alle 16 kleuren gelijktijdig op het scherm te plaatsen en tevens de helderheid van die kleuren te regelen.

In de GRAPHICS 11-mode worden de kleuren in het programma gebracht door de COLOR-instructie. Achter COLOR kunnen de getallen 0 tot en met 15 worden geplaatst waardoor de kleur van de daarna volgende tekening wordt bepaald. De getallen corresponderen met de kleurgetallen uit hoofdstuk 8.

De kleurhelderheid wordt in het programma gebracht door de SETCOLOR-instructie. Deze instructie heeft de volgende vorm: SE. 4,0,X, waarbij X de waarden 0, 2, 4, 6, 8, 10, 12 en 14 kan hebben (0=donkerste, 14=helderste tint).

Hieronder volgt een programma waarin de kleurmogelijkheden van GRAPHICS 11 worden getoond. Type en RUN:

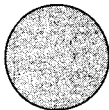
```
10 GR. 11
20 FOR A=0 TO 14 STEP 2: SE. 4,0,A
30 FOR B=0 TO 15: COLOR B
40 PL. 2*B+20,0: DR. 2*B+20,191
50 NEXT B: FOR P=1 TO 100 NEXT P: NEXT
A
60 GOTO 20
```

Door in regel 40 $2 * B + 20$ te gebruiken, worden de gekleurde strepen verticaal in het midden van het scherm geplaatst.

Taak

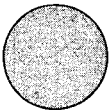
20a

Laad het programma 'Spel' van de oefencassette. RUN dit programma en voer de opdrachten uit.

Taak 20a afgerond**Taak**

20b

Laad het programma 'Terugblik' van de oefencassette. Dit is een quizprogramma over de behandelde stof van het boek.

Taak 20b afgerond

ATARI
MUSEUM

De onderstaande tabel geeft een overzicht van de SETCOLOR-, COLOR- en POKE-instructies in de grafische modes 0 tot en met 11.

	Kleur- register	SET- COLOR- instelling	POKE- adres	COLOR- instelling	Beïnvloedt de kleur van
GR. 0	1	1,-,-	709,-	-	Helderheid tekens
	2	2,-,-	710,-	-	Scherf
	4	4,-,-	712,-	-	Kader
GR. 1 - 2	0	0,-,-	708,-	-	PRINT#6, grote letters
	1	1,-,-	709,-	-	PRINT#6, kleine letters
	2	2,-,-	710,-	-	Helderheid tekst in tekstvenster PRINT#6; grote letters reverse video
	3	3,-,-	711,-	-	Tekstvenster PRINT#6; kleine letters reverse video
GR. 3 - 5 - 7	4	4,-,-	712,-	-	Achtergrond
	0	0,-,-	708,-	1	Tekeningen
	1	1,-,-	709,-	2	Tekeningen
	2	2,-,-	710,-	3	Helderheid tekst in tekstvenster Tekeningen
GR. 4 - 6	4	4,-,-	712,-	-	Tekstvenster Achtergrond
	0	0,-,-	708,-	1	Tekeningen
	1	1,-,-	709,-	-	Helderheid tekst in tekstvenster
	2	2,-,-	710,-	-	Tekstvenster
GR. 8	4	4,-,-	712,-	-	Achtergrond
	1	1,-,-	709,-	1	Helderheid tekeningen
	2	2,-,-	710,-	-	Achtergrond
GR. 9	4	4,-,0	712,-	-	Kader
	4	4,-,0	712,-	0 t/m 15	Tekeningen
	GR. 10	-	-	704,-	0
GR. 10	-	-	705,-	1	Tekeningen
	-	-	706,-	2	Tekeningen
	-	-	707,-	3	Tekeningen
	0	0,-,-	708,-	4	Tekeningen
	1	1,-,-	709,-	5	Tekeningen
	2	2,-,-	710,-	6	Tekeningen
	3	3,-,-	711,-	7	Tekeningen
	4	4,-,-	712,-	8	Tekeningen
GR. 11	-	-	-	0 t/m 15	Tekeningen (alle 16 kleuren)
	4	4,0,-	-	-	Helderheid in kleuren COLOR- instructie

Oplossingen van de opdrachten

De oplossingen die in dit aanhangsel zijn opgenomen behoeven niet gelijk te zijn aan degene die u heeft gevonden. Het is alleen belangrijk dat het programma het gewenste resultaat op het scherm brengt. Vergelijk in ieder geval uw oplossingen met de hieronderstaande oplossingen en bepaal dan wat de meest prettige manier van werken voor u is.

Het is niet de bedoeling dat alleen de in het boek opgenomen taken worden afgewerkt. Vanaf hoofdstuk drie is het aan te raden zelf programma's samen te stellen en uit te voeren.

Taak

3b

```
A$=EEN
C=38
K$=IN EEN
SOM=3001
H$=HOTEL
X=8
```

Taak

3c

U kunt de strings bijvoorbeeld A\$, B\$ en C\$ of EERSTE\$, TWEEDE\$ en DERDE\$ noemen, zolang deze namen maar gelijk zijn in de DIM-, LET- en PRINT-instructie.

```
DIM A$(10) (RETURN)
DIM B$(15) (RETURN)
DIM C$(15) (RETURN)
LET A$="IK KAN AL" (RETURN)
LET B$="OVERWEG MET " (RETURN)
LET C$="DE COMPUTER" (RETURN)
PRINT A$;B$;C$ (RETURN)
```

Elke manier is dus goed, als het resultaat maar de gevraagde zin is. (Geen spaties vergeten aan weerszijden van OVERWEG MET?)

Taak

4a

De volgorde waarin het programma wordt uitgevoerd is:
10 - 20 - 30 - 60 - 70 - 80 - 40 - 50 - 90

Op het scherm staat:

```
64=ACHT MAAL ACHT
```

De puntkomma zorgt ervoor dat regel 40 direct achter regel 70 wordt afgebeeld op het scherm.

Laat de puntkomma eens weg en bekijk dan het resultaat.

Taak

4b

```

10 PRINT"\ "
20 PRINT" |N|N|N |N|N|N |N|N|N |N|N|N "
30 PRINT" |V |B |V |B |V |B |V |B "
40 PRINT" |V| ; |B |V| , |B |V| . |B |V|P|B "
50 PRINT" |V |B |V |B |V |B |V |B "
60 PRINT" |M|M|M |M|M|M |M|M|M |M|M|M "
70 END
  
```

|N = CONTROL- plus N-toets gelijktijdig indrukken

Taak

4c

```

10 PRINT"\ "
20 PRINT" H↓←A↓←L↓←←↓←O "
  
```

↓← kan ook ←↓ zijn.

Taak

5a

Regel 70 moet zijn PRINT RUP\$

Regel 75 moet zijn PRINT"\ "

Type nu RUN

U ziet steeds ZOEK DE RUPS op het scherm verschijnen. Dit komt omdat regel 55 niet in het programma thuis hoort. Type 55 (RETURN) en regel 55 is verdwenen.

Nu ziet u iets langs flitsen. Dit wordt veroorzaakt door regel 75. Wijzig deze regel in: 40 PRINT"\ " en type RUN

Nog niet in orde. In regel 60 staan te veel cursorbewegingen. Verander regel 60 in: 60 PRINT"↓↓↓" en type RUN

De enige fout is nu nog het woord EEINDE

Verander regel 90 in 90 PRINT" EINDE "

Type RUN en het programma loopt nu goed.

Taak

6a

```

10 REM VEELVOUDEN VAN 10
20 PRINT"\ ": LET VEELV=10
30 PRINT VEELV;"+";10;"=";VEELV+10
40 VEELV=VEELV+10 :GOTO 30
  
```

De REM-instructie mag ook ergens anders staan.

De naam van de variabele (VEELV) kan alles zijn, bijvoorbeeld A. Dit is onbelangrijk, zolang het programma maar veelvoud van 10 afdruckt in een programma van vier regels.



6b

$G=10$ stopt bij $9*6$; vóór $10*6$ laat de IF...THEN-instructie het programma eindigen.

$G<10$ werkt niet. 0 is al kleiner dan 10.

$G<=10$ werkt om dezelfde reden niet.

$G>10$ is de enige oplossing die het werk doet.

$G>=10$ stopt bij $9*6$, door het = teken.

$G<>10$ werkt niet, want alle getallen behalve 10 zijn ongelijk aan 10.

U ziet dus dat het zaak is het juiste relatieteken te nemen in een IF...THEN-instructie.

In dit geval dus: $G>10$ of $G=11$ of $G>=11$.

Taak

6c

De besturingsvariabele = DOL

De startwaarde = 1

De vermeerdering = 1

De eindwaarde = 20

Taak

7a

```
PRINT 21+8-4           = 25
PRINT 4*8+2*6         = 44
PRINT (21+7*2)/(6*3-8) = 3.5
PRINT 12^2            =144
PRINT 12^2*4-16/4     =572
PRINT 2/3             = 0.6666666666
```

Taak

7b

```
10 DIM VR$(3)
20 PRINT"\ "WISSELKOERS DOLLARS"
30 PRINT"↓HOEVEEL DOLLARS WILT U WISSEL
EN"
40 INPUT DOL
50 GUL=3.72*DOL
60 PRINT:PRINT DOL;" DOLLAR=";GUL;" GUL
DEN"
70 PRINT" VERDER GAAN JA/NEE"
80 INPUT VR$
90 IF VR$="JA" THEN GOTO 20
100 IF VR$="NEE" THEN END
```

Taak

8a

```

10 PRINT"\ "
20 FOR G=1 TO 20
30 PRINT G;"+";G;"=";G+G,G;"*";G;"=";G*
G
40 NEXT G
50 END

```

Taak

8b

```

10 PRINT"\ "
20 FOR A=1 TO 8
30 PRINT "STA...";
40 FOR PAUZE=1 TO 750: NEXT PAUZE
50 PRINT"STIL!!!": PRINT: PRINT
60 FOR PAUZE =1 TO 550: NEXT PAUZE
70 NEXT A: END

```

Taak

8c

```

10 FOR A=14 TO 0 STEP -2
20 SETCOLOR 2,7,14
30 SETCOLOR 1,7,A
40 PRINT"\(hier de gekozen tekst invullen)"
50 FOR PAUZE=1 TO 500:NEXT PAUZE
60 NEXT A:END

```

Dit programma kan op verschillende manieren worden opgesteld, bijvoorbeeld door regel 20 te wijzigen in:

```
20 SETCOLOR 2,7,0
```

Dan wordt regel:

```
10 FOR A=0 TO 14 STEP 2
```

De pauze kan ook willekeurig groter of kleiner zijn.

Taak

9a

Regel 20: GETAL+3 als waarde van een variabele is fout.
 Regel 30: aanhalingstekens tussen PRINT en RESULTAAT vergeten en geen spatie voor laatste aanhalingstekens.
 Regel 40 moet zijn GOTO 10.
 Regel 50: er staat een komma tussen DATA en eerste getal.
 Regel 60 moet zijn regel 70.
 Regel 70 moet zijn regel 60.
 Geen vlag vermeld (-1); wijzig regel 20 in:
 20 IF GETAL=-1 THEN GOTO 70



9b

Voor bijvoorbeeld films en hoofdrolspelers moeten de volgende regels worden veranderd:

```

20 DIM FILM$(30), SPELER$(20), ANTW$(20)
30 READ FILM$, SPELER$
70 IF ANTW$<>SPELER$ THEN PRINT "FOUT!
HET IS "; SPELER$
90 en 100: de DATA-gegevens invullen.

```

Taak

11a

```

10 FOR Y=23 TO 0 STEP -1
20 PRINT"\ "
30 POSITION 30,Y
40 PRINT"OMHOOG"
50 FOR P=1 TO 50: NEXT P
60 NEXT Y

```

Taak

11b

```

10 FOR Y=0 TO 23
20 PRINT"\ "
30 POSITION 2, 23-Y
40 PRINT "OMLAAG"
50 POSITION 30,Y
60 PRINT "OMHOOG"
70 FOR P=1 TO 75: NEXT P
80 NEXT Y

```

U ziet, moeilijk is het niet. In één regel kunnen dus rekenkundige bewerkingen (zoals optellen of aftrekken) bij een instructie worden opgenomen. (POSITION 2, 23-Y)

Taak

11c

```

5 ? "\ "
10 FOR A=0 TO 22
20 POS. A, A
30 ? "(spatie spatie spatie)"
40 POS. A+1, A+1
50 ? "WEG"
60 FOR P=1 TO 50: NEXT P
70 NEXT A

```

Indien in regel 10 A=0 TO 23 gebruikt wordt, komt ERROR 141 op het scherm. Dit komt omdat in regel 40 de positie A+1 wordt gebruikt; hierdoor zou de laatste rij 24 worden en deze rij bestaat niet.

Taak

13a

```
10 FOR KL=0 TO 254 STEP 2
20 POKE 710, KL
30 POKE 712, KL
40 FOR P=1 TO 50: NEXT P
50 NEXT KL
```

Taak

13b

```
10 PR. "\ "
20 POS. 13, 8: PR. "VREEMDE ZAKEN"
30 FOR V=0 TO 6
40 POKE 755, V
50 FOR P=1 TO 750: NEXT P
60 NEXT V
```

Taak

14a

```
10 ? "\VOER HET AANTAL VARIABELEN IN ";
20 INPUT AANTAL
30 DIM B(AANTAL)
40 FOR K=1 TO AANTAL
50 B(K)=2*K
60 NEXT K
70 ? "B(2)+B(8)=" ; B(2)+B(8)
```

U kunt nadat het programma is afgelopen de RESET-toets indrukken en dan nogmaals het geheugen controleren door te typen:

PRINT B(X)+B(Y)

waarbij X en Y alle getallen van 1 tot en met 15 kunnen zijn.

Taak

14b

Kolom 1	Kolom 2	Kolom 3	Kolom 4
1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16
5	10	15	20
6	12	18	24
7	14	21	28
8	16	24	32
9	18	27	36
10	20	30	40

Om de waarden viermaal zo groot te maken moet regel 50 worden gewijzigd in:

```
50 M(R, K)=R*K*4
```



15a

```

1 0 ?"^\": POS. 14,12
2 0 ?"ZOEMERFEEST"
3 0 OPEN#1,4,0,"K:"
4 0 GET#1,T
5 0 IF T=84 THEN PRINT CHR$(253)
6 0 FOR A=0 TO 15
7 0 SE. 4,A,10
8 0 FOR P=1 TO 50: NEXT P
9 0 NEXT A
100 GOTO 40

```

Het stukje waar het om gaat zijn de regels 30, 40 en 50.

Regel 30 opent het toetsenbord.

In regel 40 wordt de toets ingedrukt.

Regel 50 bepaalt dat, indien de toets ASCII 84 is (de T dus), CHR\$(253) wordt geprint (dit is de zoemtoon).

Dit is een voorbeeld van ASC- en CHR\$-functies.

Taak

15b

Als het programma loopt verschijnt op het scherm:

```

0
ERROR- 13 AT LINE 90
90 NEXT F
90 NEXT X

```

moet zijn:

Type RUN en op het scherm staat nu:

```

0000000000000000
ERROR- 9 AT LINE 110

```

```

110 POS. 5,6: PRINT VLINDER$
110 POS. 5,6: PRINT VLINDER1$

```

moet zijn:

Type RUN en u krijgt:

```

ERROR-141 AT LINE 130

```

```

130 POS. 8,32: PRINT "EINDE"
130 POS. 8,19: PRINT "EINDE"

```

moet zijn:

Nu moeten de vlinderdelen nog onder elkaar komen te staan:

```

120 POS. 19,10: PRINT VLINDER2$
120 POS. 6,10: PRINT VLINDER2$

```

moet zijn:

Na het volgende RUN-commando krijgt u bovenaan:

```

0000000000000000

```

Regel 80 is de boosdoener.

```

80 PRINT NAAM moet zijn 80 PRINT NAAM$

```

In onderstaande tabel kunnen aantekeningen worden gemaakt voor de tellerstanden van de cassettebandjes. Vul de naam van het programma, de begin- en de eindstand van de teller in.

		Programmanaam		Tellerstand	
		Kant	Nr.	Aanvang	Einde
Cassette 1	A	1	Namen		
		2	Letters tikken		
		3	Vragen		
		4	Het schip		
		5	De rups		
		6	Zeg het maar		
	B	1	De lichtkrant		
		2	Kleuren		
		3	Luchtballon		
		4	Geluiden		
		5	Herhaling		
	Cassette 2	A	1	Bewegen	
2			Hoger ... lager ...		
3			Weet u het		
4			Het gemiddelde		
5			De vlinder		
6			Briefkaart		
B		1	Welke is het		
		2	Tekenen		
		3	Sorteren		
		4	Spel		
	5	Terugblik			

Foutboodschappen

De ERROR-meldingen worden op twee verschillende manieren op het scherm gebracht:

- 1 Direct, indien de fout in de getypte regel schuilt (zodat PRIT in plaats van PRINT, aanhalingstekens vergeten enz.).
- 2 Tijdens het lopen van een programma, indien door de fout het programma stopt.

Hieronder volgt een opsomming van de foutboodschappen die genummerd op het scherm verschijnen.

ERROR-

- 2 onvoldoende geheugencapaciteit (programma te lang, te veel GRAPHICS)
- 3 waarde incorrect (negatief getal waar een positief getal moet staan)
- 4 te veel variabelen (maximaal 128 in een programma)
- 5 string te lang (maximaal 3 regels)
- 6 beschikbare gegevens niet binnen bereik (geen DATA beschikbaar, op de RETURN-toets gedrukt als de cursor op READY staat)
- 7 regelnummer te groot (grootste getal is 32767)
- 8 foute INPUT-instructie (numerieke of sting-variabelen)
- 9 DIM-fout (variabele opnieuw gedimensioneerd of te grote reeks)
- 12 programmaregel niet gevonden (GOTO of GOSUB naar verkeerd regelnummer)
- 13 FOR-instructie fout (NEXT zonder FOR)
- 14 regel te lang (maximaal 114 tekens)
- 15 GOSUB- of FOR-instructie fout (GOSUB of FOR verwijderd)
- 16 RETURN-fout (wel RETURN maar geen GOSUB)
- 17 SYNTAX-fout (opdracht fout gespeld)
- 18 geen geldig string-teken (A\$ getypt als A%)
- 141 cursor buiten gestelde bereik (POS.-instructie valt buiten schermbereik)
- 144 fout extern apparaat (fout in b.v. cassetterecorder)
- 147 onvoldoende RAM-geheugen (komt voor in GRAPHICS-modes)

De volgende fouten (19 – 130 – 138) zijn INPUT/OUTPUT-fouten naar aangesloten apparatuur.

- 19 LOAD-programma te lang (verkeerd programma)
- 130 niet bestaand apparaat (apparaat niet aangesloten)
- 138 apparaat reageert niet binnen gestelde tijd (te lange aanloopstrook op cassettebandje)

BASIC-commando's, instructies en functies

In dit aanhangsel wordt een overzicht gegeven van de in dit boek behandelde commando's, instructies en functies. Achter het commando of de instructie is tussen haakjes de toegestane afkorting opgenomen.

BASIC-commando's

- BYE (B.) Zet de computer in de SELF-TEST-mode.
- CLOAD Laadt een programma vanaf de cassetteband in het geheugen van de computer.
- CONT Laat het programma vervolgen nadat de BREAK-toets is ingedrukt. Het programma gaat verder met het volgende regelnummer.
- CSAVE (CS.) Slaat een programma uit het geheugen op op een cassettebandje.
- LIST (L.) Schrijft een programma op het beeldscherm.
 LIST (het gehele programma)
 LIST 100 (alleen regel 100)
 LIST 10,200 (regel 10 tot en met regel 200)
- NEW Wist het programma uit het geheugen. **Waarschuwing:** wees voorzichtig met dit commando.
- RUN Begint de uitvoering van het programma.


BASIC-instructies

- COLOR (C.) Kiest een kleurregister in de grafische modes 3 tot en met 11. Het register is niet gelijk aan het register uit de SETCOLOR-instructie.
- DATA (D.) Bevat een lijst van gegevens die door de READ-instructie worden gelezen.
- DIM Reserveert geheugenruimte voor string-variabelen of voor numerieke reeksen.
- DRAWTO (DR.) DRAWTO X,Y trekt een lijn tussen het PLOT-punt en de coördinaten X en Y.
- END Beëindigt het programma.
- FOR (F.) Deel van de FOR...NEXT-instructie. Bepaalt het begin- en de eindwaarde van een variabele.
 Achter deze waarde kan de stap (STEP), waarmee de waarde wordt verhoogd, worden opgenomen (zonder specificering is de STEP 1).
- GET GET#1 – neemt een teken uit het gespecificeerde apparaat en zet dit in een variabele.
- GOSUB (GOS.) De uitvoering van het programma wordt vervolgd op het regelnummer achter deze instructie. De eerste RETURN-instructie laat het programma teruggaan naar de regel na de GOSUB-instructie.

- GOTO (G.)** Het programma wordt vervolgd op het regelnummer na deze instructie.
- GRAPHICS (GR.)** Kiest een grafische mode of beeldschermindeling. GRAPHICS X + 16 laat het tekstvenster verdwijnen.
- IF ... THEN** De opdracht na THEN wordt uitgevoerd als aan de voorwaarde na IF is voldaan.
- INPUT (I.)** Vraagt gegevens in te voeren. De gegevens moeten met RETURN worden afgesloten.
- LET** Kent waarden toe aan numerieke of string-variabelen.
- NEXT (N.)** Beëindigt de FOR ... NEXT-lus. Controleert of de eindwaarde is bereikt. Indien de eindwaarde is bereikt, wordt het programma vervolgd met de regel na NEXT.
- OPEN (O.)** OPEN#1,4,0,"K:" – open het toetsenbord voor een input door GET#1-bewerking.
- PLOT (PL.)** PLOT X,Y – zet een punt of teken op de door de coördinaten X en Y aangegeven plaats.
- POKE** POKE ...,X – zet het getal X (0 tot en met 255) in een bepaalde geheugenplaats.
- POSITION (POS.)** POSITION X,Y – zet de cursor op de door de coördinaten X en Y aangegeven plaats.
- PRINT (PR. of ?)** Stuurt gegevens naar het beeldscherm. Alleen PRINT geeft een lege regel.
- READ** Leest de gegevens uit de DATA-regel(s) en kent deze toe aan een variabele.
- REM (R.)** Laat een opmerking in het programma toe. De tekst achter REM wordt niet op het scherm afgedrukt.
- RESTORE (RES.)** Verwijst READ naar een DATA-regel.
- SETCOLOR (SE.)** Bepaalt de kleur en de kleurhelderheid van het gekozen kleurregister. Het registernummer is niet gelijk aan het COLOR-registernummer.
- SOUND (SO.)** Activeert een van de vier geluidskanalen om een geluid door de luidspreker van de tv te laten klinken.

Rekenkundige functies

- COS** COS(X) – geeft de cosinus van de hoek X in radialen.
- EXP** EXP(X) – geeft de waarde van e (2.718) tot de aangegeven macht.
- INT** A=INT(B) – geeft het grootste gehele getal, kleiner of gelijk aan het getal B.



RND RND(\emptyset) – geeft een willekeurig getal tussen \emptyset en 1 (nooit echter 1).
SIN SIN(X) – geeft de sinus van de hoek X in radialen.
SQR SQR(X) – geeft de vierkantswortel van X.

Numerieke functies

FRE FRE(\emptyset) – geeft het aantal vrije geheugenplaatsen.
PEEK PEEK(X) – geeft het getal dat in het geheugenadres X is opgeslagen.

String-functies

ASC ASC("A") – geeft de ATASCII-code van het teken A.
CHR\$ CHR\$(X) – geeft het teken dat door ASC X wordt vertegenwoordigd.
LEN LEN(A\$) – geeft een getal dat de lengte van de string A\$ aangeeft.


































De ATASCII-code - de CHR\$-code - de schermcode










ATASCII vormt een afkorting van ATARI ASCII. De letters en de cijfers hebben dezelfde waarde als in ASCII, maar enkele speciale karakters zijn verschillend.


De karakters 128 tot en met 255 zijn de reverse video afbeeldingen van de karakters 1 tot en met 127.




Om een lettercode voor kleine letters te krijgen moet bij de code van de hoofdletter 32 worden opgeteld.

Om het karakter van de CHR\$-functie op het scherm te krijgen, moet het getal van de ATASCII-code tussen haakjes achter CHR\$ worden geplaatst: CHR\$(...).

ATASCII-code	CHR\$-functie	ATASCII-code	CHR\$-functie	ATASCII-code	CHR\$-functie
0		34		68	D
1		35	"	69	E
2		36	\$	70	F
3		37	%	71	G
4		38	&	72	H
5		39	'	73	I
6		40	(74	J
7		41)	75	K
8		42	*	76	L
9		43	+	77	M
10		44	'	78	N
11		45	-	79	O
12		46	.	80	P
13		47	/	81	Q
14		48	0	82	R
15		49	1	83	S
16		50	2	84	T
17		51	3	85	U
18		52	4	86	V
19		53	5	87	W
20		54	6	88	X
21		55	7	89	Y
22		56	8	90	Z
23		57	9	91	[
24		58	:	92	\
25		59	;	93]
26		60	<	94	^
27		61	=	95	_
28		62	>	96	
29		63	?	97	a
30		64	@	98	b
31		65	A	99	c
32	spatie	66	B	100	d
33	!	67	C	101	e

ATASCII-code	CHR\$-functie	ATASCII-code	CHR\$-functie	ATASCII-code	CHR\$-functie
102	f	146		190	
103	g	147		191	
104	h	148		192	
105	i	149		193	
106	j	150		194	
107	k	151		195	
108	l	152		196	
109	m	153		197	
110	n	154		198	
111	o	155		199	
112	p	156		200	
113	q	157		201	
114	r	158		202	
115	s	159		203	
116	t	160		204	
117	u	161		205	
118	v	162		206	
119	w	163		207	
120	x	164		208	
121	y	165		209	
122		166		210	
123		167		211	
124		168		212	
125		169		213	
126		170		214	
127		171		215	
128		172		216	
129		173		217	
130		174		218	
131		175		219	
132		176		220	
133		177		221	
134		178		222	
135		179		223	
136		180		224	
137		181		225	
138		182		226	
139		183		227	
140		184		228	
141		185		229	
142		186		230	
143		187		231	
144		188		232	
145		189		233	



ATASCII-code	CHR\$-functie	ATASCII-code	CHR\$-functie
234		245	
235		246	
236		247	
237		248	
238		249	
239		250	
240		251	
241		252	
242		253	 (<i>zoemer</i>)
243		254	 (<i>Delete</i>)
244		255	 (<i>Insert</i>)

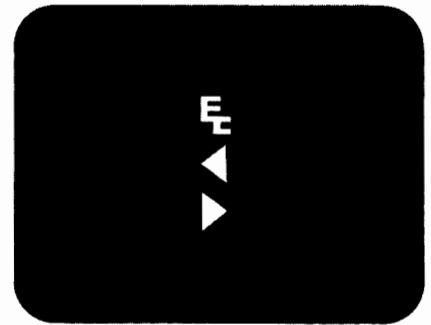
Indien de onderstaande toetsen in de aangegeven volgorde worden ingedrukt, verschijnen op het scherm, tussen de aanhalingstekens van de PRINT-instructie, de daarnaast aangegeven beelden.

De functie van de toetsen (het scherm schoonmaken, de cursorbewegingen e.d.) treedt pas in werking als het programma loopt.

Druk op



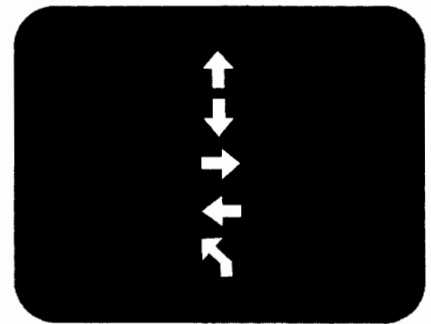
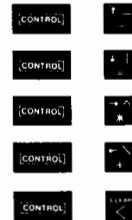
Druk op



Druk op



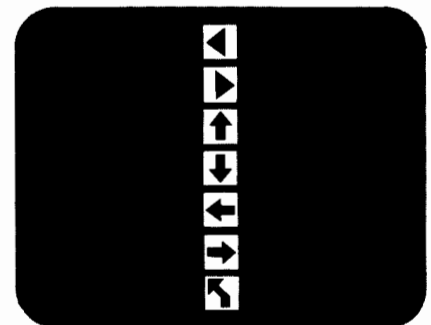
Druk samen in



Druk op



Druk samen in



Belangrijke POKE- en PEEK-adressen

- ATARI
MUSEUM.NL
- POKE 16,64 schakelt de BREAK-toets uit.
- POKE 20=PEEK(20) geeft de waarde van de ingebouwde timer.
- POKE 82,X bepaalt de linker kantlijn (X=0 tot en met 39).
- POKE 83,X bepaalt de rechter kantlijn (X=0 tot en met 39).
- POKE 201,X bepaalt de breedte van de PRINT-tabulator (normaal 10).
- POKE 559,0 schakelt het scherm uit voor snellere bewerkingen.
- POKE 580,1 laat het systeem opnieuw starten bij het indrukken van de RESET-toets.
- POKE 708,X geeft kleurregister 0 de kleur X.
- POKE 709,X geeft kleurregister 1 de kleur X.
- POKE 710,X geeft kleurregister 2 de kleur X.
- POKE 711,X geeft kleurregister 3 de kleur X.
- POKE 712,X geeft kleurregister 4 de kleur X.
- POKE 752,1 schakelt de cursor uit (752,0 zet hem weer aan).
- POKE 755,0 schakelt reverse video en cursor uit.
- POKE 755,1 wist alle tekst die in reverse video staat en schakelt de cursor uit.
- POKE 755,4 drukt alle tekst onderste boven af zonder cursor.
- POKE 755,6 drukt alle tekst onderste boven af met cursor.
- POKE 756,204 geeft in combinatie met de CONTROL-toets toegang tot de internationale karakterset.
- POKE 756,224 geeft hoofdletters.
- POKE 756,226 geeft kleine letters.
- PEEK(763) geeft het laatste ASCII-karakter dat werd gelezen of geschreven.
- PEEK(764) geeft de code van de laatste ingedrukte toets.
- POKE 767,0 geeft de normale controle van START/STOP met de CONTROL- en 1-toets.
- POKE 767,1 stopt het omhoogrollen van de tekst.
- POKE 53279,0 laat de luidspreker klinken (pieptoon).
- PEEK(53279) geeft aan of een functietoets wordt ingedrukt: 3=OPTION-toets, 5=SELECT-toets en 6=START-toets.

PEEK(53775) geeft 251 als een toets wordt ingedrukt, 247 als de SHIFT-toets wordt ingedrukt en 255 als geen toets wordt ingedrukt.

POKE 54018,52 schakelt de motor van de cassetterecorder in.

POKE 54018,60 schakelt de motor van de cassetterecorder uit.

Omzettingstabel

SETCOLOR naar POKE

SETCOLOR kleurregister, kleur, helderheid = POKE geheugenadres, waarde. Bijvoorbeeld bij SETCOLOR 2,0,0 = POKE 710,0; dit is SE. achtergrond(2), grijs(0), helderheid(0) = POKE achtergrond(710), waarde(0).

Op de plaats van # wordt het kleurregister en het POKE-geheugenadres ingevuld.

kleur 0 (grijs)		kleur 3 (rood/oranje)		kleur 6 (paars/blauw)	
SE.#,0,0 =	POKE #,0	SE.#,3,0 =	POKE #,48	SE.#,6,0 =	POKE #,96
2 =	2	2 =	50	2 =	98
4 =	4	4 =	52	4 =	100
6 =	6	6 =	54	6 =	102
8 =	8	8 =	56	8 =	104
10 =	10	10 =	58	10 =	106
12 =	12	12 =	60	12 =	108
14 =	14	14 =	62	14 =	110

kleur 1 (goud)		kleur 4 (roze)		kleur 7 (blauw)	
SE.#,1,0 =	POKE #,16	SE.#,4,0 =	POKE #,64	SE.#,7,0 =	POKE #,112
2 =	18	2 =	66	2 =	114
4 =	20	4 =	68	4 =	116
6 =	22	6 =	70	6 =	118
8 =	24	8 =	72	8 =	120
10 =	26	10 =	74	10 =	122
12 =	28	12 =	76	12 =	124
14 =	30	14 =	78	14 =	126

kleur 2 (oranje)		kleur 5 (roze/paars)		kleur 8 (blauw)	
SE.#,2,0 =	POKE #,32	SE.#,5,0 =	POKE #,80	SE.#,8,0 =	POKE #,128
2 =	34	2 =	82	2 =	130
4 =	36	4 =	84	4 =	132
6 =	38	6 =	86	6 =	134
8 =	40	8 =	88	8 =	136
10 =	42	10 =	90	10 =	138
12 =	44	12 =	92	12 =	140
14 =	46	14 =	94	14 =	142

kleur 9 (lichtblauw)

SE.#,9,0 =	POKE #,144
2 =	146
4 =	148
6 =	150
8 =	152
10 =	154
12 =	156
14 =	158

kleur 12 (groen)

SE.#,12,0 =	POKE #,192
2 =	194
4 =	196
6 =	198
8 =	200
10 =	202
12 =	204
14 =	206

kleur 14 (oranje/groen)

SE.#,14,0 =	POKE #,224
2 =	226
4 =	228
6 =	230
8 =	232
10 =	234
12 =	236
14 =	238

kleur 10 (turquoise)

SE.#,10,0 =	POKE #,160
2 =	162
4 =	164
6 =	166
8 =	168
10 =	170
12 =	172
14 =	174

kleur 13 (geel/groen)

SE.#,13,0 =	POKE #,208
2 =	210
4 =	212
6 =	214
8 =	216
10 =	218
12 =	220
14 =	222

kleur 15 (lichtoranje)

SE.#,15,0 =	POKE #,240
2 =	242
4 =	244
6 =	246
8 =	248
10 =	250
12 =	252
14 =	254

kleur 11 (groen/blauw)

SE.#,11,0 =	POKE #,176
2 =	178
4 =	180
6 =	182
8 =	184
10 =	186
12 =	188
14 =	190

Geluid

In dit aanslagel zijn de beschikbare tonen opgenomen, met daarachter het getal dat in de DATA-regel moet worden geplaatst. Daarna volgen enige geluidseffecten. Deze geluiden kunnen in programma's worden verwerkt waarin geluid belangrijk is (bijvoorbeeld spelletjes).

G-balk

F-balk

The image shows two musical staves. The top staff is labeled 'G-balk' and has a treble clef. The bottom staff is labeled 'F-balk' and has a bass clef. Both staves show a sequence of notes: C, D, E, F, G, A, B, C. The notes are placed on the lines and spaces of the staves to represent their pitch.

HOGE TONEN	C	29
	B	31
	A# OF B ^b	33
	A	35
	G# OF A ^b	37
	G	40
	F# OF G ^b	42
	F	45
	E	47
	D# OF E ^b	50
	D	53
	C# OF D ^b	57
	C	60
	B	64
	A# OF B ^b	68
	A	72
	G# OF A ^b	76
	G	81
	F# OF G ^b	85
F	91	
E	96	
D# OF E ^b	102	
D	108	
C# OF D ^b	114	
MIDDEN C	C	121
	B	128
	A# OF B ^b	136
	A	144
	G# OF A ^b	153
	G	162
F# OF G ^b	173	
F	182	
LAGE TONEN	E	193
	D# OF E ^b	204
	D	217
	C# OF D ^b	230
	C	243

Sommige van de hieronder opgenomen geluidseffecten zijn eenmalig, dat wil zeggen dat als u ze meerdere malen achter elkaar wilt horen, u een GOTO- of FOR...NEXT-instructie aan het programma moet toevoegen.

```

5 REM VOGEL
10 FOR K=1 TO 10
20 FOR TH=1 TO 28
30 SOUND 2,TH,10,8
40 NEXT TH: NEXT K

```

```

5 REM VALLEND VOORWERP
10 FOR TH=30 TO 200
20 SOUND 2,TH,10,8
30 NEXT TH

```

```

5 REM ONWEER
10 FOR A=1 TO 5
20 FOR TH=1 TO 255
30 SOUND 0,TH,8,15
40 NEXT TH
50 FOR P=1 TO 500:NEXT P
60 NEXT A

```

```

5 REM STOOMLOCOMOTIEF
10 FOR V=15 TO 0 STEP -1
20 SOUND 1,15,0,V
30 NEXT V: GOTO 10

```

```

5 REM SIRENE
10 FOR TH =100 TO 10 STEP-1
20 SOUND 0,TH,10,9: NEXT TH
30 FOR TH=10 TO 100
40 SOUND 0,TH,10,9
50 NEXT TH: GOTO 10

```

```

5 REM ZOEMER
10 SOUND 2,40,6,10
20 FOR P=1 TO 400: NEXT P

```

Stoppen van verschillende van bovenstaande programma's moet worden gedaan door de RESET-toets in te drukken of in het programma een END-instructie in te bouwen.

Register

- Aaneenschakelen
 - PRINT-instructie 31
- Aansluiten 10
- Achtergrondkleur 68
- Afkortingen 91
- Afstellen tv 11
- Aftrekken 58
- Array's 107
- ASC-functie 114, 170

- Besturingsvariabele 54
- Bewegende beelden 90
- Beweging 90
- Bewerkingsvolgorde 58
- BYE-commando 12, 68

- Cassetterecorder 13, 46
- CHR\$-functie 114, 170
- COLOR-instructie 136, 154, 168
- Commando's 26, 167
- CONT-commando 39, 168
- COS-functie 146, 169
- Cursorbeweging 21
 - in PRINT-instructie 41

- DATA-instructie 74, 168
- Decimale punt 27
- Delen 58
- DIM-instructie 30, 168
- Dollarteken 29
- DRAWTO-instructie 137, 168
- Dubbele punt 51

- Eindroutine 124
- Eindwaarde 54
- E-notatie 94

- ERROR- 14, 166

- FOR...NEXT-instructie 64, 168
- Foutboodschappen 14, 166
- FRE-functie 97, 170
- Functies 94, 169

- Geheugen RAM 28
- Geheugen ROM 100
- Geluid 79
- Geluidseffecten 84, 183
- Geluidsinstelling 80, 182
- Geneste lussen 63
- Getallen 93
- GET-instructie 116, 168
- GOSUB-instructie 120, 168
- GOTO-instructie 38, 169
- Grafische symbolen 22, 33, 40
- GRAPHICS 67, 127, 154

- Haakjes 59
- Helderheid toon 82
- Hoogte toon 82, 182

- IF...THEN-instructie 52, 169
- INPUT-instructie 60, 169
- Inschakelen 11
- Instructies 25, 167
- INT-functie 95, 169

- Kaderkleur 68
- Kleine letters 20, 32
- Kleur 67
- Kleuren in GRAPHICS 130, 154
- Kleurentabel 68
- Kleuregister 130, 154

- ATARI
MUSEUM.NL
- Lengete noot 81, 104
 - Lengete PRINT-instructie 33
 - LEN-functie 116, 170
 - LET-instructie 28, 169
 - LIST-commando 38, 168
 - LOAD-commando 14, 168

 - Matrix 110

 - NEW-commando 32, 168
 - NEXT 64, 169
 - Noot, lengte 81, 104
 - Numerieke functies 94, 170
 - Numerieke variabelen 29

 - OPEN-instructie 116, 169
 - Oplossingen taken 155
 - Ontwerpen programma's 120
 - Optellen 58

 - Pauze-lus 65
 - PEEK-functie 103, 170
 - PLOT-instructie 137, 169
 - POKE-instructie 101, 169
 - POSITION-instructie 88, 169
 - PRINT-instructie 26, 169
 - Programma 36, 119
 - Punt-komma 31

 - RAM-geheugen 28
 - READ-instructie 74, 169
 - Relatietekens 53
 - REM-instructie 50, 169
 - RESTORE-instructie 77, 169
 - RETURN-instructie 121, 169
 - Reverse video 21, 33
 - RND-functie 95, 170
 - ROM-geheugen 100
 - RUN-commando 14, 168

 - SAVE-commando 46, 168
 - Schermcodes 171
 - Scherm schoonmaken 19
 - SETCOLOR-instructie 68, 169
 - SIN-functie 146, 170
 - SOUND-instructie 82, 169
 - Speciale toetsen 19
 - SQR-functie 94, 170
 - Startwaarde 54
 - STEP 64
 - String-functies 113, 170
 - String-variabelen 29
 - Subroutines 120
 - Supervisor 120

 - Taken, oplossingen 155
 - Tekens, kleur 68
 - Tellerstanden 163
 - Testen computer 12
 - THEN 52
 - Toetsenbord 18
 - Toon, helderheid 82
 - Toon, hoogte 82, 182

 - Variabelen 28
 - Verbeteren fouten 44
 - Vermeerdering 54
 - Vermenigvuldigen 58
 - Vlag 74
 - Volume 82
 - Voorwaardelijke instructies 52

 - Wiskundige notatie 95

LEER IN BASIC PROGRAMMEREN OP DE ATARI

Auteurs: J. F. Hellwig sr. en
J. F. Hellwig jr.

Een zelfstudie pakket voor beginners die met de ATARI BASIC willen leren programmeren.

Deze cursus is speciaal ontwikkeld voor diegenen die geen voorkennis van het programmeren of het werken met de ATARI-computer bezitten.

De behandelde stof is geschikt voor de meeste ATARI-computers van zowel de 400/600/600XL/800XL/65XE/130XE serie. Uitvoerig komen de mogelijkheden die de ATARI-computer biedt aan de orde. De cursus omvat:

- het leren gebruiken van de vele grafische mogelijkheden van de ATARI-computers;
- het opstellen van

- geluids- en muziekprogramma's;
 - het, aan de hand van duidelijke voorbeelden, opstellen van programma's voor het opslaan van gegevens;
 - het maken van berekeningen, spelletjes e.d.;
 - het opsporen van fouten in slecht lopende programma's.
- Het cursusboek en de programma's op de cassettes zijn geheel op elkaar afgestemd. Op de cassettes zijn programma's opgenomen die of oefeningen bevatten waarin de behandelde stof wordt herhaald, of voorbeelden die in het boek uitvoerig worden toegelicht.



Inhoud:

- cursusboek
- twee cassettes met tweeëntwintig programma's

ISBN 9001 97036 2

Wolters
SOTWOTTE