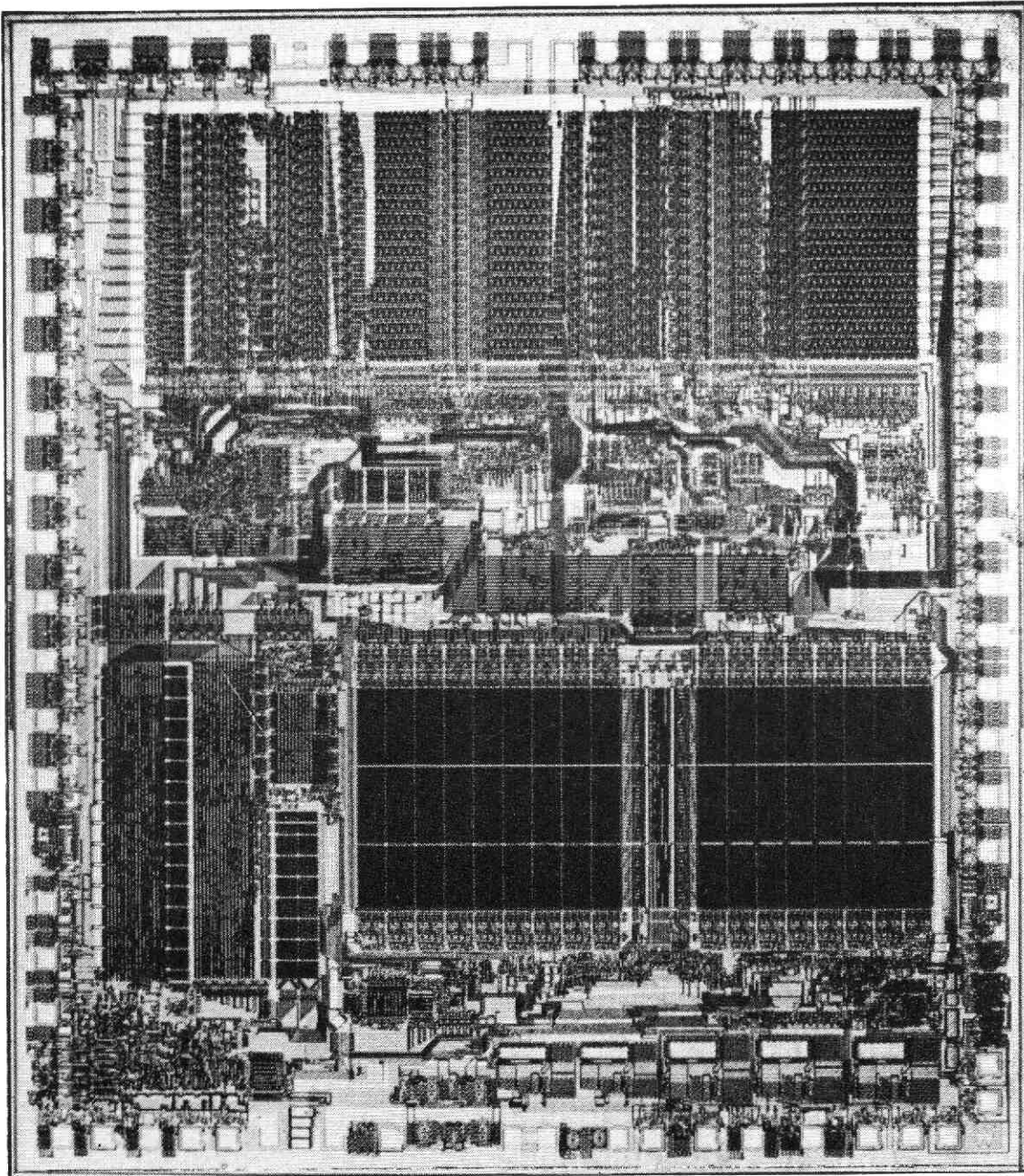




# Automatische gegevensverwerking

<https://hdl.handle.net/1874/10257>



# AUTOMATISCHE GEGEVENS - VERWERKING



Freudenthal instituut  
Archief

**AUTOMATISCHE  
GEGEVENS-  
VERWERKING**



Tiberdreef 4 - 3561 GG Utrecht

*Foto omslag:*

*Een microprocessor, het hart van een computer.*

AUTOMATISCHE GEGEVENSVERWERKING

Een produktie ten behoeve van de experimenten in het kader van de Herverkaveling Eindexamenprogramma's Wiskunde I en II V.W.O.

Samenstelling: Heleen Verhage  
Guis Vonk

Vormgeving: Ellen Hanepen

© 1984; 3e herziene versie

Utrecht, juni 1984.

INHOUDSOPGAVE

1	Algoritmen en structuurdiagrammen	1
2	Programmeren in Basic of Comal	21

# 1

## ALGORITMEN EN STRUCTUURDIAGRAMMEN

- » 1. Een scholier werkt één avond in de week in een disco en verdient daarmee in 1980 100 gulden per avond. Met de eigenaar is afgesproken dat hij de komende jaren dit bedrag zal blijven verdienen, maar dat het voor inflatie gecorrigeerd zal worden. In 1981 verdient André dan ook 106 gulden, want de inflatie was 6%. In '1980 guldens' is dit nog steeds 100 gulden. Welk bedrag moet André in 1982 krijgen om nog steeds 100 "1980 guldens" te verdienen? En in 1986? (Ga er bij de berekening vanuit dat de inflatie 6% per jaar blijft).

Als je veel moet uitrekenen op de rekenmachine, loont het de moeite om dit zo handig mogelijk te doen.

Je kunt het loon in 1981 als volgt berekenen:

$$\text{loon 1981} = 6\% \text{ van loon 1980} + \text{loon 1980}.$$

Maar het kan ook zó:

$$\text{loon 1981} = 1.06 \times \text{loon 1980}.$$

Dit laatste rekt sneller, ga maar na!

In dit hoofdstuk gaan we een nieuwe notatie invoeren om de bewerkingen voor je rekenmachine (en later voor de computer) ondubbelzinnig op te schrijven.

Bij het uitrekenen van het loon in 1986 op je rekenmachine ben je waarschijnlijk ook de lonen voor de tussenliggende jaren tegengekomen.

Volgens de handige rekenmethode:

$$\text{loon } 1982 = 1.06 \times \text{loon } 1981$$

$$\text{loon } 1983 = 1.06 \times \text{loon } 1982$$

⋮

$$\text{loon } 1986 = 1.06 \times \text{loon } 1985.$$

Zo zou je elk tussenstapje in de berekening met een formule kunnen beschrijven, om aan te geven wat je gedaan hebt.

Gelukkig kan het korter:

Je kunt 'loon' opvatten als een *variabele* die voor elk jaar een andere waarde aanneemt (die kan variëren).

Zo'n variabele heeft een *naam* en een *waarde*. In dit voorbeeld krijgt de variabele met de naam *loon* voor het jaar 1980 de waarde 100.

Dit wordt genoteerd als:

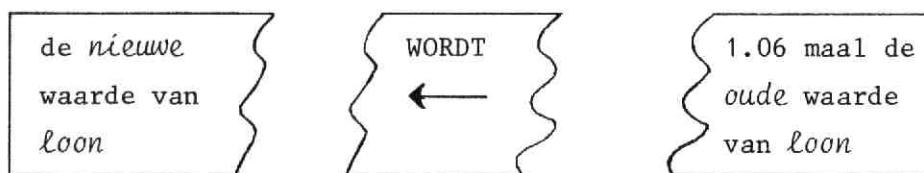
$$\text{loon} \leftarrow 100$$

en uitgesproken als: 'de waarde van loon *wordt* 100.'

Voor de jaren ná 1980 geldt:

$$\text{loon} \leftarrow 1.06 * \text{loon}$$

Je moet dit begrijpen als:

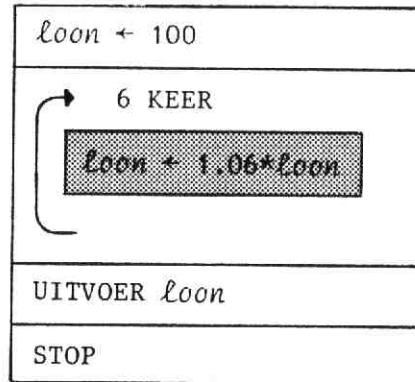


Aan de variabele *loon* wordt dus een *nieuwe* waarde toegekend.

Deze *nieuwe* waarde ontstaat door de *oude* waarde met 1.06 te vermenigvuldigen.

- De vermenigvuldiging wordt aangegeven met een sterretje (\*), dit om verwarring met de letter x te voorkomen. We laten het vermenigvuldigingsteken *niet* weg.
- Als decimaalteken wordt een punt (.) gebruikt, zoals dat in de Angelsaksische landen de gewoonte is, dus géén komma (,).

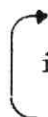
- » 2. De berekening van André's loon in 1986 staat in het volgende schema:



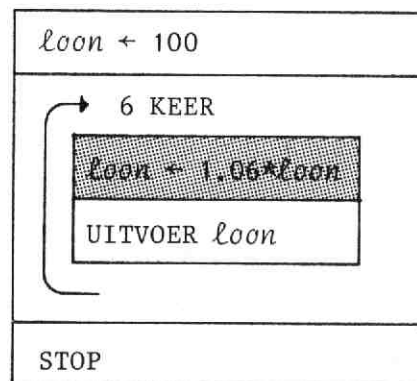
UITVOER $loon$
----------------

wil zeggen dat de waarde van  $loon$  opgeschreven moet worden.

Zo'n schema heet een *structuurdiagram*.

- a. Wat zal de pijl  in het structuurdiagram betekenen?

Een andere mogelijkheid is:



- b. Wat is het verschil in verwerking tussen de twee structuurdiagrammen?
- c. Welk structuurdiagram past het beste bij de berekening die je bij opgave » 1 hebt uitgevoerd?

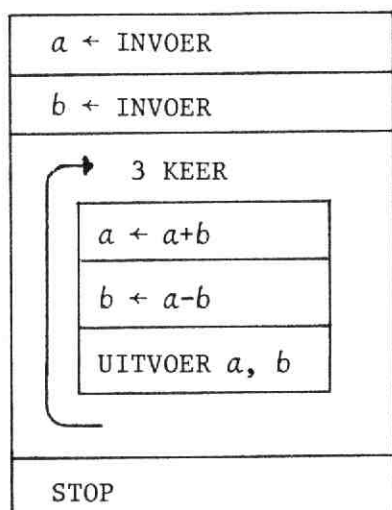


- » 3. a. Neem aan dat de koopkracht van ambtenaren in het midden van de jaren 80 met 3% per jaar afneemt. Stel de koopkracht in 1982 op 100 en maak een structuurdiagram waarin aangegeven staat hoe de koopkracht voor de volgende vier jaar berekend kan worden.
- b. Voer de opdrachten uit dit structuurdiagram stap voor stap uit om de koopkracht voor ambtenaren uit te rekenen.

$A \leftarrow \text{INVOER}$
------------------------------

betekent dat de variabele A een nader op te geven waarde krijgt.

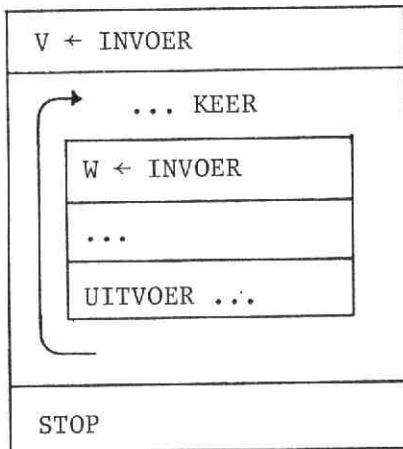
- » 4. Neem als invoer voor het onderstaande structuurdiagram 2 en 1. Doorloop het diagram en noteer de uitvoer in een tabel.



$a$	$b$
2	1
..	..
..	..
..	..

- » 5. Een verkoopster van wasmachines voorspelt aan het eind van iedere maand het aantal wasmachines dat de volgende maand verkocht zal worden. Als voorspelling voor de nieuwe maand neemt zij het gemiddelde van de voorspelling ( $V$ ) van de vorige maand en de werkelijke verkopen ( $W$ ) van de vorige maand.
- Ze voorspelt dat in januari 100 wasmachines verkocht zullen worden. Het werkelijke aantal verkochte wasmachines in januari bedraagt 102.
- a. Bereken de voorspelling voor februari.

- b. De invoer van het structuurdiagram hieronder bestaat uit de voorspelling voor januari en de werkelijke verkopen over januari tot en met december. De uitvoer moet zijn: de voorspellingen voor februari t/m januari van het volgend jaar zoals de verkoopster die berekend heeft.



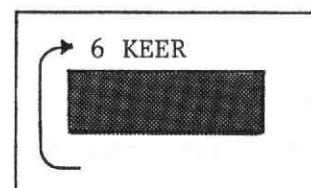
maand	werkelijke verkopen	voorspelling
jan	102	100
febr	104	...
maart	104	...
april	108	...
mei	114	...
juni	116	...
juli	118	...
aug	124	...
sept	124	...
okt	128	...
nov	124	...
dec	130	...

- c. Doorloop het structuurdiagram met als invoer de gegevens uit de tabel.
- d. Wat vind je van de voorspelmethode van de verkoopster?

Structuurdiagrammen worden opgebouwd uit een aantal blokken, waarin vastgelegd is wat er moet gebeuren als het diagram doorlopen wordt. Met behulp van speciale blokken wordt aangegeven *hoe* het diagram doorlopen moet worden.

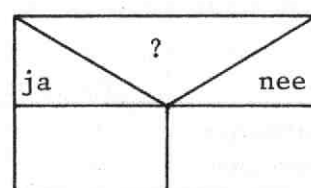
In de vorige opgaven is al gebruik gemaakt van het *herhalingsblok*:

Van te voren staat vast hoe vaak de opdrachten in het gearceerde stuk herhaald moeten worden.

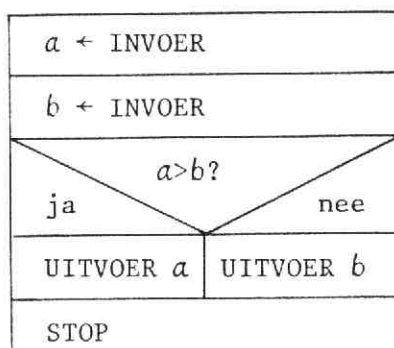


Een ander speciaal blok is het *keuzeblok*:

De bewering op de plaats van het vraagteken kan waar en niet-waar zijn. Afhankelijk daarvan wordt gekozen welke opdracht vervolgens uitgevoerd moet worden.



- » 6. a. Neem twee willekeurige waarden voor a en b en kijk wat het resultaat van dit structuurdiagram is:



- b. Verander het structuurdiagram zó, dat het grootste min het kleinste getal uitgevoerd wordt.



17e eeuwse gevecht met een  
'meenichte' beren op  
Spitsbergen.

Op het eiland Spitsbergen was in de 17e eeuw de nederzetting Smeerenburg gevestigd, van waaruit op walvissen gejaagd werd. Het spek van de walvissen werd ter plekke tot traan verkookt.

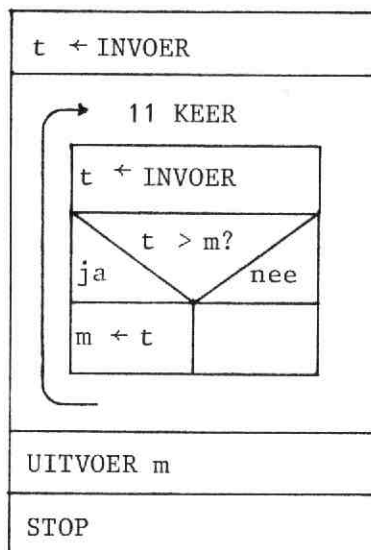
Dit alles was mogelijk dankzij de warme golfstroom, die er voor zorgt dat de temperatuur op Spitsbergen ( $78^\circ$  NB) niet al te ijzig is.

Een van de koudste plaatsen op aarde is het Siberische Werchojansk ( $68^\circ$  NB). Het heeft een heel ander klimaat dan Spitsbergen.

Kijk maar naar de volgende tabel:

	jan	feb	mrt	apr	mei	juni	juli	aug	sep	okt	nov	dec
Spitsbergen	-11	-11	-12	-9	-4	2	5	4	2	-4	-8	-10
Werchojansk	-50	-40	-30	-12	2	13	15	11	-4	-15	-37	-47

» 7.



- a. Wat is in dit structuurdiagram de uitvoer als de invoer de temperatuur op Spitsbergen is?
- b. En als de invoer de temperatuur op Werchojansk is?
- c. Omschrijf in woorden wat dit structuurdiagram doet.

» 8. a. Maak een structuurdiagram om de gemiddelde jaartemperatuur uit te rekenen.

Hint: maak gebruik van een variabele *som*, die aan het begin de waarde nul krijgt en vervolgens steeds gelijk gemaakt wordt aan de som van de al ingevoerde temperaturen, volgens:

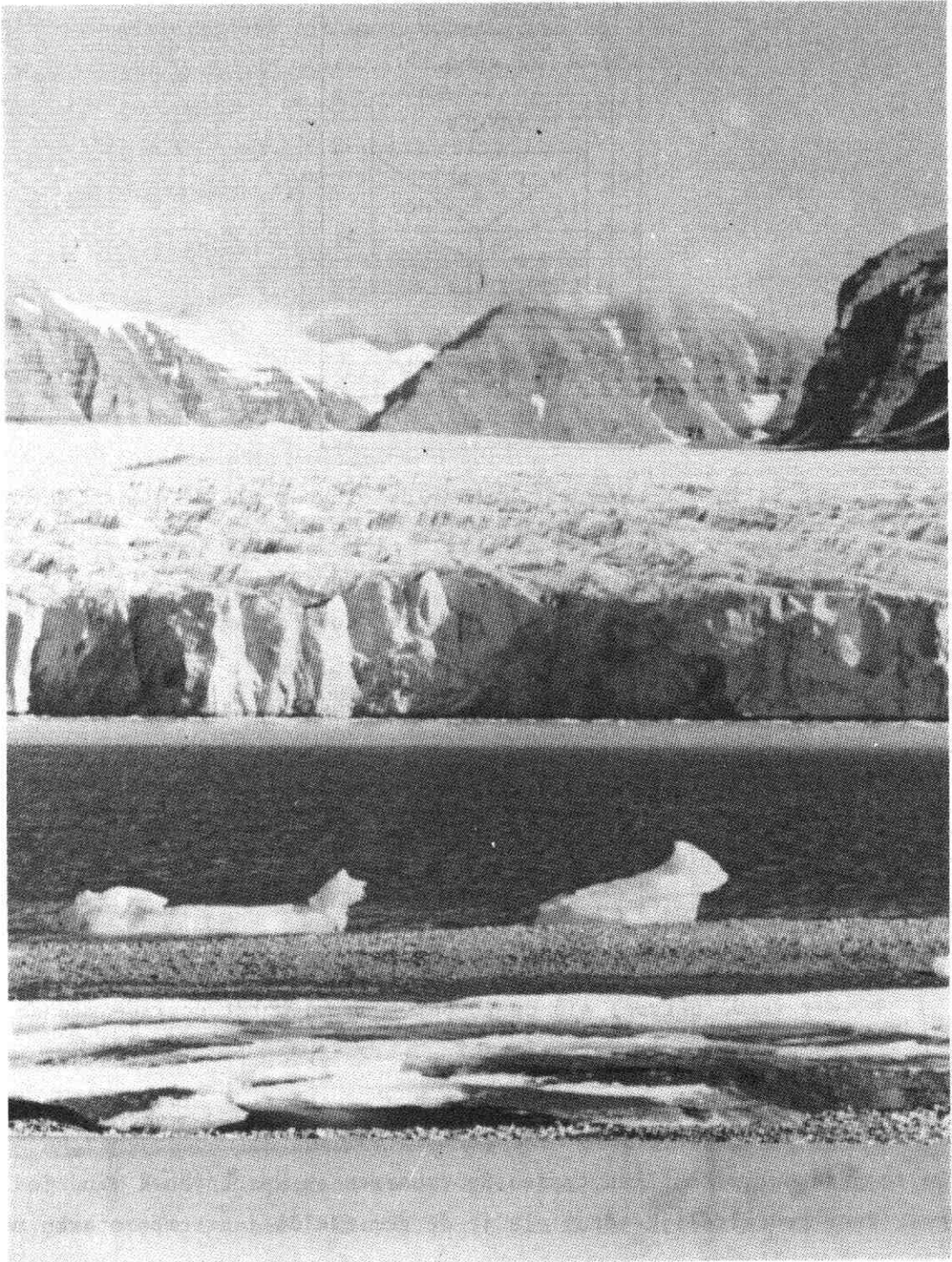
$\text{som} \leftarrow \text{som} + \text{temp}$
--

- b. Wat is de uitvoer als de invoer gevormd wordt door de maandtemperaturen op Werchojansk?

De gemiddelde jaartemperatuur op Spitsbergen is  $-4.7^{\circ} \text{C}$ .

De temperatuurverschillen tussen Spitsbergen en Werchojansk vind je maar zeer gedeeltelijk terug als je de gemiddelde jaartemperaturen met elkaar vergelijkt.

Daarom gaan we nu voor beide gebieden afzonderlijk na in welke mate de maandtemperaturen afwijken van de gemiddelde jaartemperatuur.



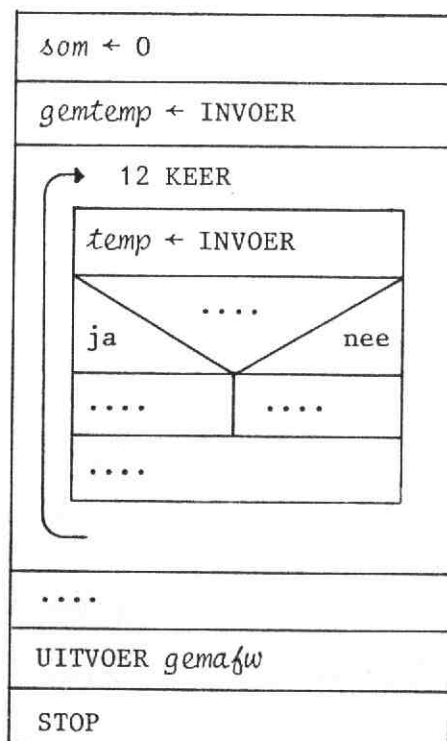
*De Posbreengletsjer stroomt in de Tempelfjord  
op Spitsbergen ver in zee.*

We doen dit als volgt:

- Neem voor elke maand het verschil tussen de maandtemperatuur en de jaartemperatuur. Laat hierbij het teken buiten beschouwing (dus voor januari op Spitsbergen wordt het 6,3 in plaats van - 6,3).
- Tel al deze verschillen bij elkaar op.
- Neem het gemiddelde.

Het getal dat je zo vindt, wordt wel de *gemiddelde absolute afwijking* genoemd.

- » 9. a. Maak het onderstaande structuurdiagram om de gemiddelde absolute afwijking te berekenen verder af.



- b. Bereken met je rekenmachine de gemiddelde absolute afwijking van de maandtemperaturen op Spitsbergen. De gemiddelde absolute afwijking van de maandtemperatuur in Werchojansk gaan we in het volgende hoofdstuk met behulp van de computer berekenen.

In het voorafgaande zijn onder andere voorschriften beschreven om:

- het loon in opeenvolgende jaren te berekenen;
- de vraag naar wasmachines te voorspellen;
- het gemiddelde te bepalen.

Een dergelijk voorschrift, dat precies aangeeft wat er moet gebeuren om tot het verlangde resultaat te komen, wordt een *algoritme* \*) genoemd.

Algoritmen kent iedereen eigenlijk allang, ook al worden ze niet altijd met die naam aangegeven. Bijvoorbeeld 'het optellen van twee getallen met elk veel cijfers'. Of het vermenigvuldigen ervan. Of het maken van een staartdeling. Maar ook verrichtingen die minder met rekenen te maken hebben, zoals een breipatroon, een kookrecept of de aanwijzingen in een telefooncel.

Een algoritme heeft o.a. de volgende eigenschappen:

- ze kan worden voltooid in een eindig aantal stappen;
- na elke stap staat ondubbelzinnig vast wat de volgende stap is of dat de algoritme eindigt.

We beschrijven algoritmen ondermeer met behulp van structuurdiagrammen.

---

### *franse lelies*

Aantal st. deelbaar door 6, plus 3 st.  
**1e en 3e nld.** : r.: 3 st. blauw, \* 1 st. wit, 5 st. blauw \*.  
**2e nld.** : av.: 1 st. wit, \* 3 st. blauw, 3 st. wit \*, 2 st. blauw.  
**4e en 6e nld.** : av.: 2 st. blauw, \* 1 st. wit, 5 st. blauw \*,  
 1 st. wit.  
**5e nld.** : r.: 2 st. wit, \* 3 st. blauw, 3 st. wit \*, 1 st. blauw.



\*) In de negende eeuw schreef Abu Gafar Mohammed ibn Musa in Bagdad een boek over de Indische rekenkunst. Abu Gafar werd wel *al hwarizmi* genoemd: de uit de landstreek Hwarizm afkomstige. Van al hwarizmi stamt het woord algoritme.

## ALGORITMEN OP DE REKENMACHINE

Tonia Maat, liefhebster van onbespoten groente en fruit, wil een flinke lap grond geschikt maken om er zelf gewassen op te verbouwen.

Ze heeft berekend dat ze  $200 \text{ m}^2$  nodig heeft.

De moestuin moet rechthoekig worden en bovendien wil To zo min mogelijk materiaal kwijt zijn aan de omheining.

In een opwelling tekent ze eerst eens een rechthoekig veld met *lengte* 20 en *breedte* 10. De omtrek wordt dus 60 meter.

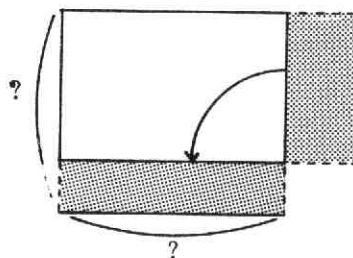
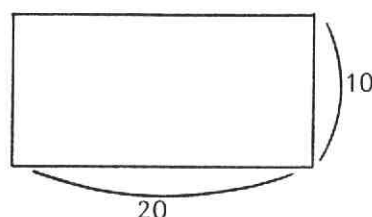
Ze is nog niet erg tevreden over het resultaat en vermoedt dat het beter kan door de lengte wat te verkleinen en de breedte wat te vergroten.

Voor de *nieuwe lengte* neemt ze het gemiddelde van de *oude lengte* en de *breedte*:

$$\text{lengte} + (\text{lengte} + \text{breedte})/2.$$

Omdat de oppervlakte 200 moet zijn, is hiermee ook de *nieuwe breedte* bepaald:

$$\text{breedte} + 200/\text{lengte}.$$



» 10. a. Wat wordt de *nieuwe lengte*?

Bereken de *breedte* die daarbij hoort tot in centimeters nauwkeurig.

Ga na dat de omtrek inderdaad is afgenomen.

b. Schrijf je antwoorden van a) in dit schema en herhaal het proces nog twee keer.

Hoe zou je het *geheugen* van de rekenmachine bij deze berekening kunnen benutten?

<i>lengte</i>	<i>breedte</i>
20	100
..	..
..	..
..	..

c. Welke vorm krijgt het tuintje uiteindelijk?

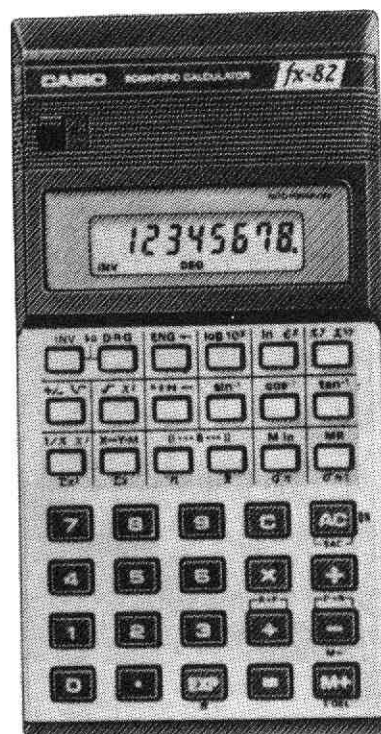
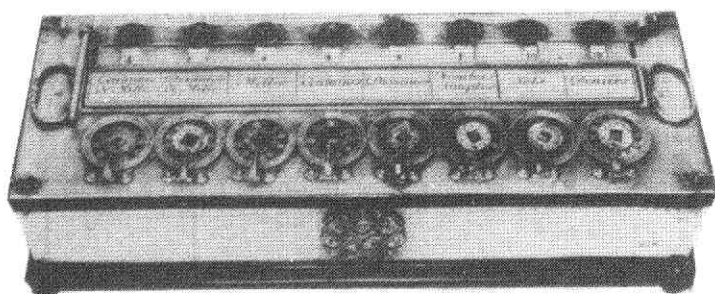
d. Maak een structuurdiagram waarin beschreven staat hoe deze berekening verlopen is.



De methode die To gebruikt om de maten van haar tuintje te bepalen, is in feite een handige manier om  $\sqrt{200}$  te benaderen. Als je de tussenresultaten niet afrond, is het resultaat na vier keer al tot vier decimalen nauwkeurig (14.1421). Je kunt de nauwkeurigheid verder vergroten door het proces nog vaker te herhalen.

#### HET GEHEUGEN VAN DE REKENMACHINE

Als je bij de vorige opgave het geheugen van de rekenmachine niet gebruikt hebt, moest de waarde van lengte steeds tweemaal worden ingetoest: voor het berekenen van de nieuwe *breedte* en voor het berekenen van de nieuwe *lengte*.



Links de rekenmachine van Pascal (1642). Bij deze machine stellen de posities van de wielen de getallen voor. Op deze machine kunnen alleen maar gehele getallen worden verwerkt. Rechts een wat eigentijds(er) apparaat.

Door het *geheugen* (memory) van de rekenmachine te gebruiken kun je je het werk dat dit met zich meebrengt besparen.

In het geheugen kan een getal worden opgeslagen dat op dat moment niet nodig is, maar later in dezelfde berekening wèl.

De meeste rekenmachines hebben een knop om het getal van het venster in het geheugen op te slaan. Als er een opgeslagen waarde in het geheugen staat, kun je gewoon verder rekenen met het venster. De inhoud van het geheugen kan op elk gewenst moment weer teruggeroepen worden naar het venster.

De knoppen die gebruikt moeten worden om met het geheugen te werken, hangen af van het merk rekenmachine.

De volgende aanduidingen komen nog al eens voor:

Min of  STO Slaat het getal uit het venster op in het geheugen.

MR of  RCL De inhoud van het geheugen verschijnt in het venster.

AC Maakt het venster schoon.

MC ,  CM • Maakt het geheugen schoon.

M+ Telt het getal uit het venster op bij het getal in het geheugen.

M- Trekt het getal uit het venster af van het getal in het geheugen.

» 11. Vul in welke knoppen je op jouw rekenmachine moet gebruiken om het volgende te bewerkstelligen:

*geheugen* + *venster* is

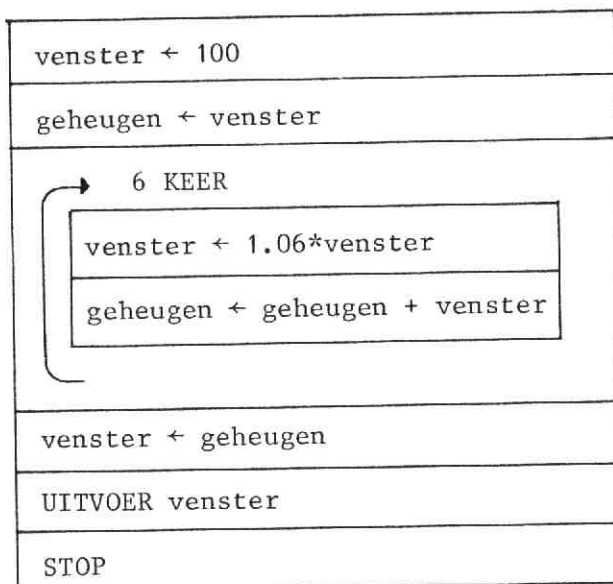
*geheugen* + 0 is

*venster* + *geheugen* is

*geheugen* + *geheugen* + *venster* is

*venster* + *venster* + *geheugen* is

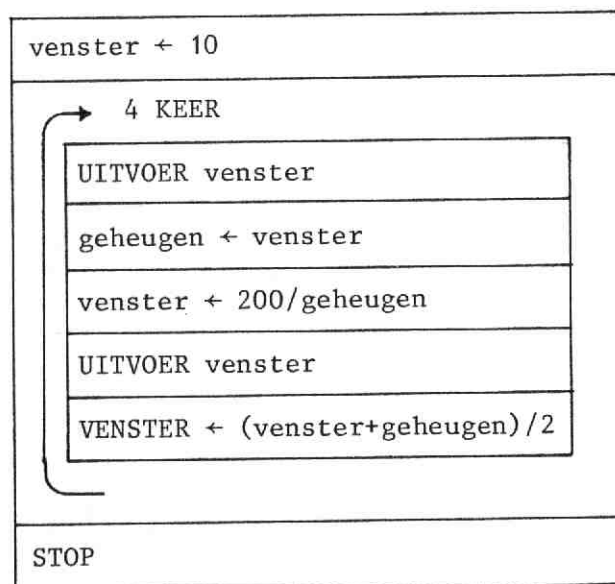
» 12. a. Wat is de uitvoer bij het doorlopen van het volgende structuurdiagram?



b. Bedenk een probleem waar dit structuurdiagram het antwoord op geeft.

We gaan het geheugen van de rekenmachine nu gebruiken om  $\sqrt{200}$  te benaderen.

Om aan te geven hoe je het *uitleesvenster* en het *geheugen* van je rekenmachine bij deze opgave handig kunt benutten, passen we het structuurdiagram wat aan.



De variabelen *lengte* en *breedte* komen nu niet meer voor. In plaats daarvan wordt het voorschrift nu uitgedrukt in de variabelen *venster* en *geheugen*.

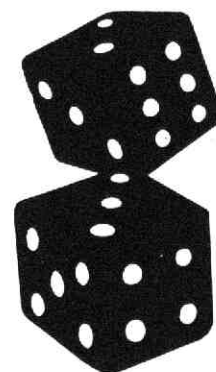
- » 13. Voer de berekening om  $\sqrt{200}$  te benaderen uit en benut daarbij het geheugen van de rekenmachine zoals aangegeven in het structuurdiagram.

Controleer het antwoord met de  $\sqrt{\quad}$ -knop.

- » 14. Kijk nog eens naar opgave » 5.

Maak nu een structuurdiagram waar uit blijkt hoe de rekenmachine handig gebruikt kan worden.

Gebruik alleen de variabelen *venster* en *geheugen*.



#### TOEVALSGETALLEN

Bij de kansrekening wordt gebruik gemaakt van toevalsgetallen.

#### TOEVALSGETALLEN

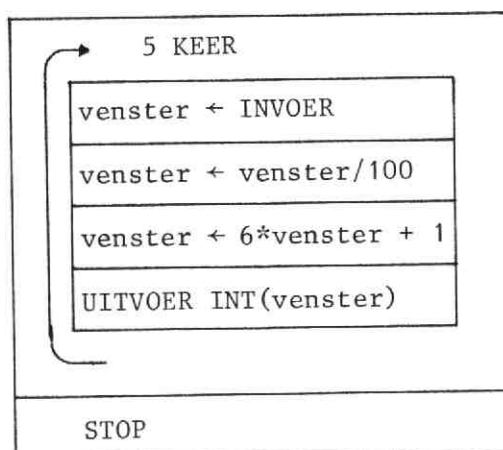
71	08	73	19	01	84	69	70	20	24	46	51	46	05	57	53	52	53	68	76	11	71	92	70	88
23	06	60	38	85	70	24	60	04	24	27	33	26	23	53	60	86	10	61	66	56	94	20	20	09
35	51	67	74	11	11	53	51	43	59	62	34	22	73	87	19	73	83	19	59	23	37	19	91	78
50	87	96	18	11	59	44	77	84	26	49	39	69	19	14	09	19	24	86	14	42	47	91	47	85
46	53	63	70	04	92	44	87	65	75	59	46	67	41	81	01	19	58	30	70	48	96	62	77	03
52	17	65	68	35	90	11	61	22							85	89	17	49	21	03	80	49	32	11
13	17	79	34	88	09	37									28	23	89	77		55	48	79	01	81
30	63	23	43	97	11										35	34				40	52	37	50	36
90	48	28	56	25																42	70	24		
22	24	55																						

Toevalsgetallen kun je gebruiken om een toevalsexperiment na te bootsen (te *simuleren*).

» 15. In het volgende structuurdiagram is beschreven hoe je met behulp van toevalsgetallen en een rekenmachine het gooien van een dobbelsteen kunt simuleren.

De opdracht  $\text{INT}(\text{venster})$  betekent 'het gehele deel nemen van de waarde van *venster*. Dus bijvoorbeeld  $\text{INT}(5.26)$  geeft als uitkomst 5.

(INT is een afkorting van integer, Engels voor 'geheel'.)



- Wat is de uitvoer als de invoer bestaat uit de eerste vijf getallen uit de tabel met toevalsgetallen?
- Wat moet er in het structuurdiagram veranderen als het gooien van een vierkante dobbelsteen met de getallen 2, 3, 4 en 5 op de zijden gesimuleerd moet worden?

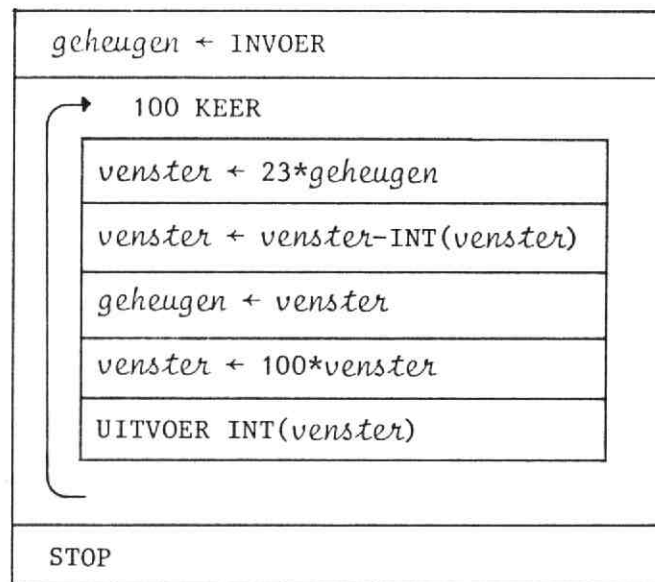
Je kunt je afvragen hoe zo'n bladzijde toevalsgetallen tot standkomt. Hoe 'toevallig' zijn die getallen eigenlijk?

Een manier om aan toevalsgetallen te komen is, om ze te *maken* op de rekenmachine. Dit klinkt gek, want een rekenmachine kan toch niets toevallig doen?

Dat is ook zo en die getallen die je krijgt worden in feite berekend volgens een zeker voorschrift. Ze worden daarom *pseudo-toevalsgetallen* genoemd. Het gaat er nu om een voorschrift te kiezen dat aan de volgende voorwaarden voldoet:

- alle getallen van twee cijfers moeten ongeveer even vaak voorkomen;
- als de rij getallen die ontstaat zich eventueel gaat herhalen, dan mag dat in elk geval niet te snel gebeuren.

Het voorschrift dat in dit structuurdiagram beschreven is, voldoet aan deze voorwaarden:



Als je het voorschrift voor de eerste keer uitvoert met als INVOER 0.439147, krijg je de volgende tussenresultaten:

```

geheugen ← 0.439147
10.100381 ← 23 * 0.439147
 0.100381 ← 10.100381 - INT(10.100381)
geheugen ← 0.100381
10.0381 ← 100 * 10.100381
10 opschrijven

```

- » 16. a. Ga dit na en bereken de volgende drie pseudotoevalsgetallen.  
 b. Krijg je ook pseudotoevalsgetallen als je als startgetal 0.500000 neemt in plaats van 0.439147?

Kennelijk geeft het voorschrift alléén bij een geschikte keuze van het startgetal een methode om pseudo-toevalsgetallen te maken. Maar dan herhaalt de rij zich ook pas na 25000 getallen!

- » 17. a. Verander het structuurdiagram van opgave » 16 zodanig dat het geschikt is om 10 vierkeuzevragen op de gok te beantwoorden.
- b. Doorloop het structuurdiagram en neem als invoer een startgetal dat aan de volgende voorwaarden voldoet:
- het ligt tussen 0 en 1 en heeft 6 decimalen;
  - het laatste cijfer is oneven en ongelijk aan 5.
- (Bruikbaar is dus bijv. 0.833547, onbruikbaar is 0.833545).

Het voorschrift om pseudotoevalsgetallen te maken hoeft voor computers meestal niet uitgeschreven te worden. Met de opdracht:

getal ← RANDOM
----------------

wordt aangegeven dat de variabele *getal* een willekeurige waarde in  $[0,1>$  krijgt. (Random: Engels voor lukraak.)

» 18.

a ← INVOER
b ← INVOER
getal ← RANDOM
getal ← INT(a*getal+b)
UITVOER getal

- a. Wat moet de invoer zijn als de uitvoer een willekeurig geheel getal uit  $\{0, 1, \dots, 99\}$  moet zijn?
- b. En als de uitvoer uit  $\{1, 2, \dots, 6\}$  moet zijn?

Gebruik de RANDOM-functie bij de volgende opgaven:

- » 19. Ontwerp een algoritme om 1000 keer tossen met een eerlijke munt te simuleren. Hierbij moet geturfd worden. Dit tellen doen we door iedere keer dat 'kop' gegooid is

$$kop \leftarrow kop + 1$$

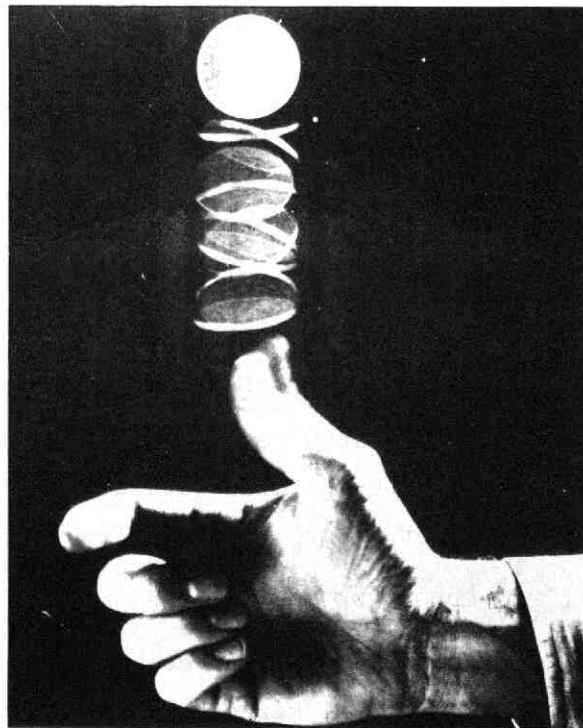
te verwerken. Vergeet niet bovenaan de beschrijving  $kop \leftarrow 0$ . UITVOER moet zijn het aantal keren kop en het aantal keren munt.

» 20. Een dronkeman heeft bij iedere stap die hij doet een kans  $2/3$  om een pas voorwaarts te maken en een kans  $1/3$  om een pas in tegengestelde richting te maken.

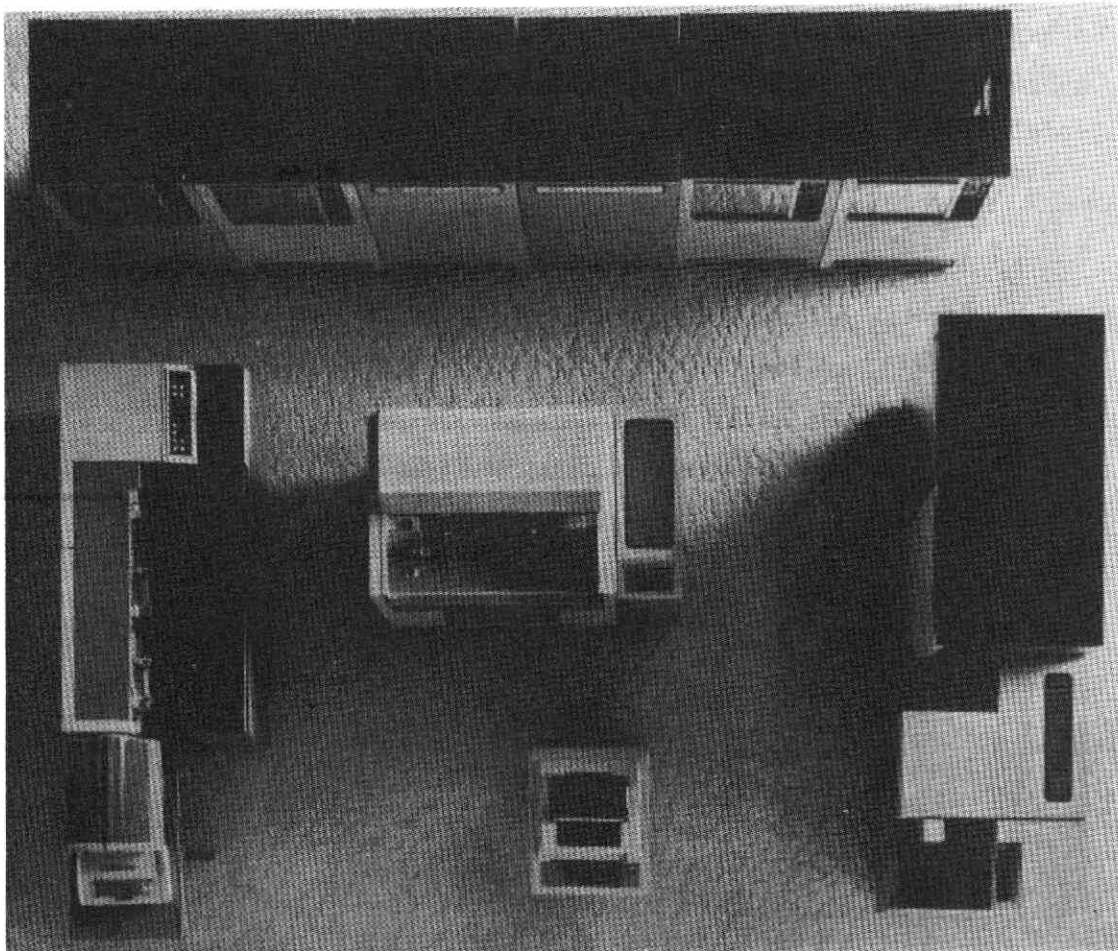
a. Beschrijf een simulatie van 10 stappen waarbij bepaald wordt hoeveel hij uiteindelijk in de goede richting gevorderd is.

b. Hoever is de denkbeeldige dronkeman gevorderd als de RANDOM-functie achtereenvolgens oplevert:

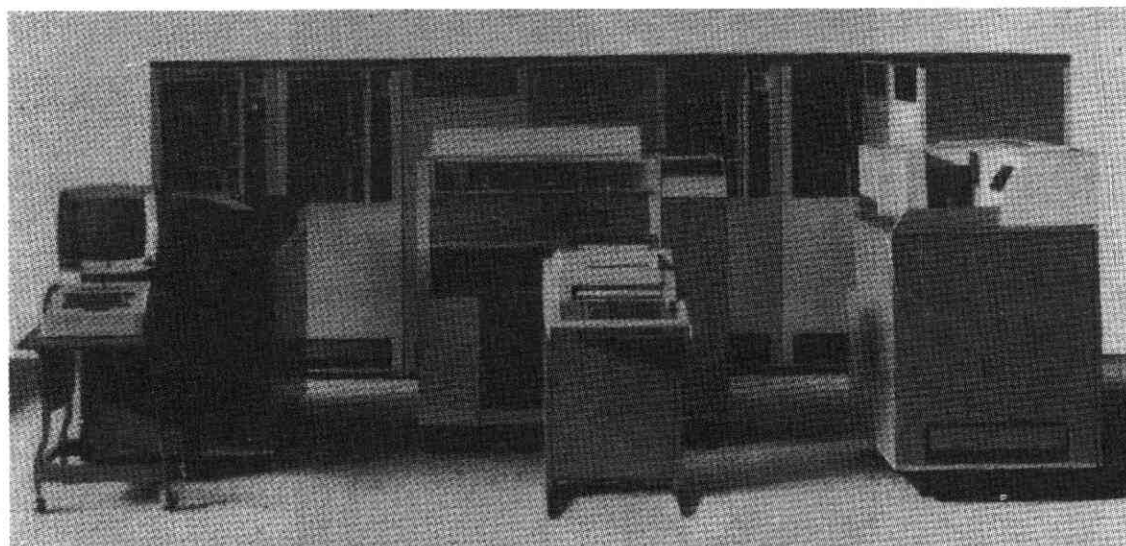
0.486203	0.765353
0.781923	0.596781
0.119288	0.872262
0.705801	0.464657
0.198359	0.758792







*Een grote computer.*



# 2

## PROGRAMMEREN IN BASIC OF COMAL



We gaan nu een computer inschakelen voor het verwerken van onze algoritmen.

Als een algoritme beschreven is in een structuurdiagram, is het uiterlijk nog niet geschikt voor een computer. Deze machine is niet in staat onze schema's te bekijken. Het diagram moet omgezet worden in een *computerprogramma*. Een programma is de beschrijving van een algoritme in een vorm die bestemd is voor een computer en liefst ook nog leesbaar voor de mens. Dit laatste veronderstelt dat er gebruik gemaakt wordt van een soort taal, de *programmeertaal*.

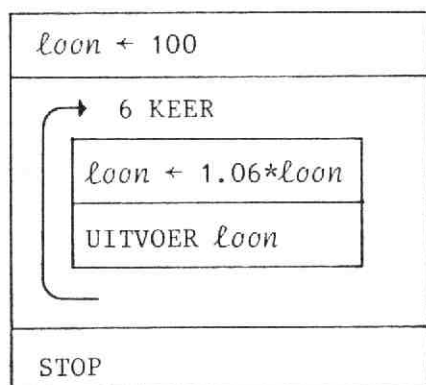
Er zijn heel wat verschillende programmeertalen ontwikkeld. Dit hoofdstuk is gebaseerd op de taal BASIC (Beginners' All-purpose Symbolic Instruction Code). De woorden die BASIC kent zijn zoals dat bij veel programmeertalen het geval is, in het Engels.

Dit hoofdstuk is ook bruikbaar als met COMAL (COMon Algorithmic Language) gewerkt wordt.

De wijzigingen t.o.v. BASIC staan in voetnoten vermeld.

Als het computerprogramma geschreven is, kun je het vervolgens door de computer laten *verwerken*. Computers zijn er in alle soorten en maten. In dit hoofdstuk wordt gesproken over de verwerking door middel van *micro-computers*.

Het structuurdiagram van opgave » 2 kan zonder al teveel moeite omgezet worden in een Basic-programma: \*)

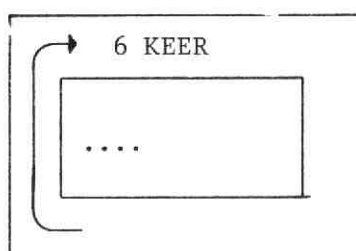


```

10 L = 100
20 FOR T = 1 TO 6
30 L = 1.06 * L
40 PRINT L
50 NEXT T
60 END

```

Het herhalingsblok wordt vertaald met de FOR-NEXT opdracht:



*structuurdiagram*

```

FOR T = 1 TO 6
...
NEXT T

```

*computerprogramma*

De variabele T is een teller die achtereenvolgens de waarden 1, 2, 3, 4, 5 en 6 krijgt.

» 21. Bekijk ook goed hoe de andere bestanddelen uit het structuurdiagram vertaald worden.

De programmatekst kan nu op een toetsenbord worden ingetypt.

\*) Comal:

Het teken  $\leftarrow$  wordt naar COMAL vertaald met :=

Regel 20 moet vervangen worden door: 20 FOR T := 1 TO 6 DO.

## AANWIJZINGEN BIJ HET GEBRUIK VAN EEN MICROCOMPUTER

- Laat een keer voordoen hoe de machine aangezet en gestart wordt.
- Wat wel en niet kan bij het intypen en verwerken van een programma hangt sterk af van de gebruikte microcomputer.
- De regels van het programma zijn stijgend genummerd, met sprongen van 10 omhoog. Het voordeel hiervan is, dat later nog regels kunnen worden tussengeschoven.
- De namen van variabelen worden aangegeven met behulp van hoofdletters. (Soms zijn alleen namen van één letter toegestaan, bijv. L i.p.v. LOON).
- Spaties worden verkregen door de spatiebalk in te drukken.
- De toets RETURN (ook wel ENTER ) dient om van het einde van een regel naar de volgende regel te komen.
- De nul (0) verschijnt op het scherm met een streepje erdoor: Ø, dit om verwarring met de letter o te voorkomen.

### *Wat te doen bij fouten?*

- Je merkt de fout terwijl je nog op de regel aan het typen bent. Ga met de toets BACK SPACE of ← (afhankelijk van de machine) terug naar de plaats des onheils en typ vanaf die plaats de juiste tekst.
- Een regel blijkt bij nader inzien onjuist. Typ de regel helemaal opnieuw in met hetzelfde regelnummer. De computer zorgt er zelf voor dat de regel op de juiste plaats in het programma komt te staan.
- De computer weigert een regel achteraf met een of andere foutmelding. Druk op RETURN en typ de regel opnieuw in.

### *Als je klaar bent met intypen:*

- Met LIST RETURN krijg je een opsomming (lijst) van wat je aan programma hebt ingetypt. Dit woord is geen deel van het programma en heeft daarom geen voorafgaand nummer.
- Met RUN RETURN kun je een programma verwerken.
- Met NEW RETURN kun je een programma verwijderen.  
Pas gebruiken als je zeker weet dat je het programma niet meer nodig hebt.
- Met BREAK wordt de verwerking voortijdig afgebroken. (Ook wel DEL of ESC )

» 22. a. Typ het programma van opgave » 21 en verwerk het.

Komen de resultaten overeen met je antwoorden op opgave » 1?

$loon \leftarrow \text{INVOER}$

wordt in BASIC: INPUT L

b. Vervang regel 10 op deze wijze en verwerk het programma nog eens.

De regel met INPUT L zal bij verwerking van het programma zorgen dat om een waarde gevraagd wordt. Je ziet op het scherm een vraagteken. Typ een getal gevolgd door `RETURN`. Dat wordt de waarde van *loon*.

Je kunt met de PRINT-opdracht ook *tekst* laten afdrucken.

Deze tekst moet tussen aanhalingstekens staan. Bijvoorbeeld:

```
PRINT "Loon berekenen"
```

geeft als uitvoer:

```
Loon berekenen
```

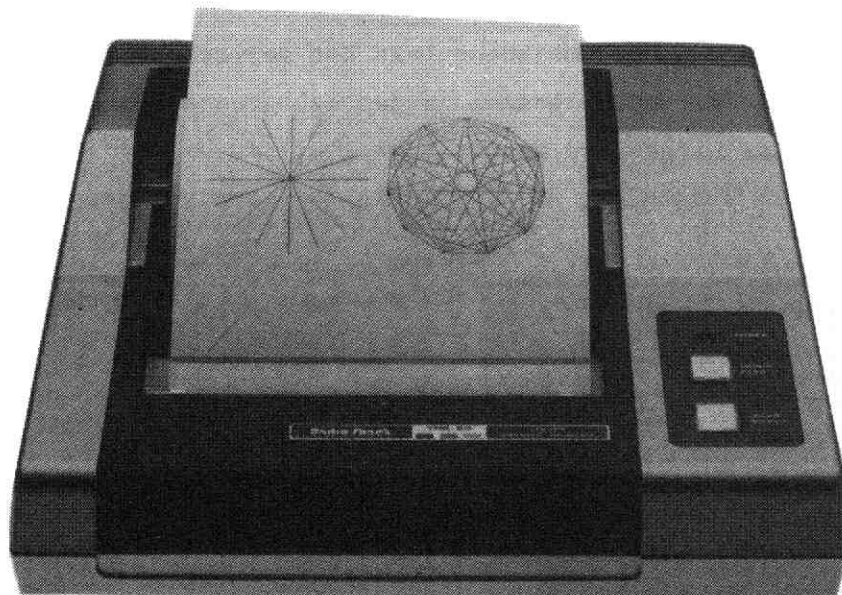
Door geschikte PRINT "tekst" opdrachten in een programma op te nemen, kun je er voor zorgen dat bij de verwerking van een programma duidelijk is waar het programma voor dient. De *gebruikersvriendelijkheid* van het programma is dan vergroot.

Het programma van opgave » 21 zou je bijvoorbeeld als volgt uit kunnen breiden:

```
5 PRINT "GEEF HET LOON IN HET BASISJAAR (JAAR 0) OP"
10 INPUT L
20 FOR T=1 TO 6
30 L = 1.06*L
40 PRINT "Het LOON IN JAAR"; T; "is: ";L
50 NEXT T
60 END
```

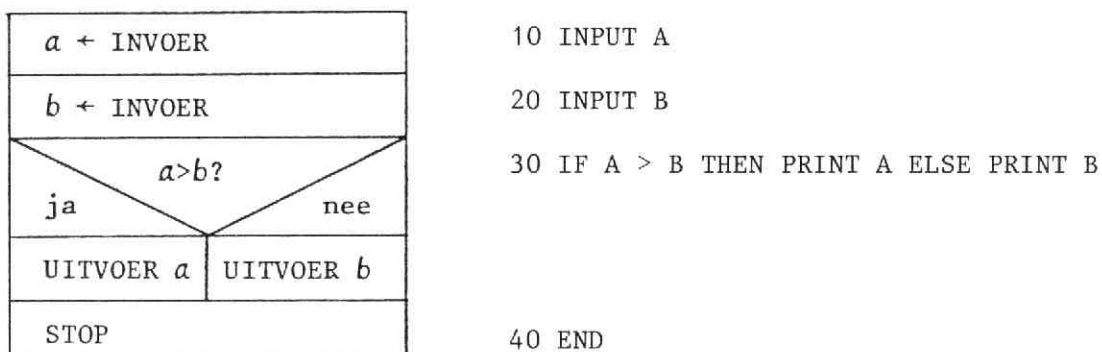
Uit regel 40 blijkt, dat met één PRINT-opdracht zowel tekst als variabelen afgedrukt kunnen worden. De punt-komma (;) doet dienst als scheidingsteken in de PRINT-opdracht.

- » 23. a. Verander het programma in deze zin en verwerk het nog eens.  
b. Verander het programma zodanig dat het inflatiepercentage pas bij verwerking opgegeven hoeft te worden.
- » 24. Vertaal het structuurdiagram van opgave » 3 in een programma en verwerk het op de computer.
- » 25. a. Maak een computerprogramma bij opgave » 4.  
b. Verander het programma zodanig dat:  
- de herhaling 20 keer doorlopen wordt;  
- het quotiënt van a en b ook wordt afgedrukt.
- » 26. Verwerk opgave » 5 op de computer.  
Er moet een tabel uitgevoerd worden, waarin maandnummer, werkelijke verkopen en voorspelde verkopen naast elkaar staan.

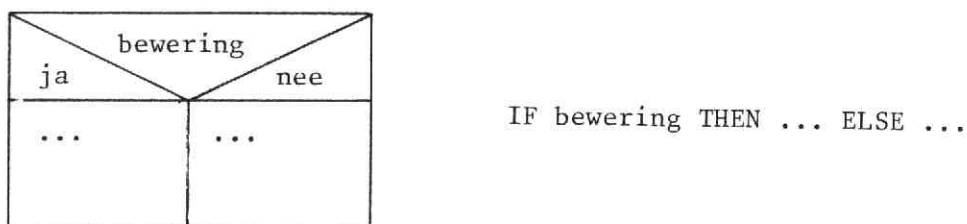


*Grafische printer*

Het structuurdiagram van opgave » 6 wordt zó vertaald in een programma: \*)



Het keuze-blok wordt vertaald met de IF-THEN-ELSE opdracht:



*structuurdiagram*

*computerprogramma*

- » 27. a. Typ het programma in en laat het verwerken.  
 b. Verander het programma zo dat het grootste min het kleinste getal uitgevoerd wordt.

\*) In sommige Basic-varianten komt de IF-THEN-ELSE opdracht niet voor, maar alleen IF-THEN. Regel 30 moet dan vervangen worden door (met gebruikmaking van de sprongopdracht goto):

```
30 IF A > B THEN PRINT A : GOTO 40
35 PRINT B
```

De dubbele punt (:) op regel 30 is een scheidingsteken tussen BASIC-opdrachten.

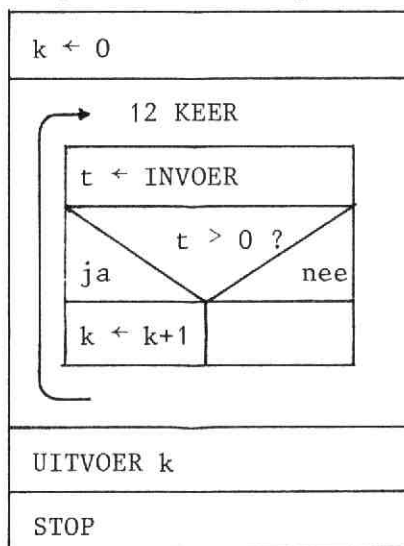
\*) Comal:

```
Regel 30 vervangen door: 30 IF A > B THEN
                        40 PRINT A
                        50 ELSE
                        60 PRINT B
                        70 ENDIF
```

De maandtemperaturen voor Spitsbergen en Werchojansk waren:

	jan	feb	mrt	apr	mei	juni	juli	aug	sep	okt	nov	dec
Spitsbergen	-11	-11	-12	-9	-4	2	5	4	2	-4	-8	-10
Werchojansk	-50	-40	-30	-12	2	13	15	11	-4	-15	-37	-47

» 28. a. Wat doet het volgende structuurdiagram als de invoer bestaat uit de maandtemperaturen van Spitsbergen?



b. Vertaal dit structuurdiagram in een computerprogramma en verwerk het met als invoer de maandtemperaturen van Werchojansk.

» 29. Schrijf met behulp van het structuurdiagram van opgave » 8 een programma dat de gemiddelde jaartemperatuur bepaalt.

De invoer bestaat uit 12 maandcijfers.

Verwerk het programma met als INVOER de gemiddelde maandtemperatuur van Spitsbergen.



*Het leven op Spitsbergen  
in de 17e eeuw.*



» 30. Met het volgende programma kan de gemiddelde absolute afwijking berekend worden. Wat stellen de variabelen G, T, V, S en A voor?\*)

```

10 INPUT G
20 S=0
30 FOR N=1 TO 12
40 INPUT T
50 IF T>G THEN V=T-G ELSE V=G-T
60 S=S+V
70 NEXT N
80 A=S/12
90 PRINT A
100 END

```

Typ het programma in en verwerk het voor de maandtemperaturen van Werchojansk. Vergelijk de uitkomst met je antwoord op opgave » 9.

Op regel 50 staat in feite niets anders dan 'neem de absolute waarde van het verschil van T en G'. Je kunt V ook rechtstreeks berekenen door gebruik te maken van de ABS-functie en regel 50 te vervangen door:

```
50 V = ABS(T - G)
```

» 31. Het discowezen bloeit enorm in de jaren '80, met als gunstig gevolg voor André dat hij bovenop zijn salaris in 1980 van 100 gulden per avond jaarlijks tien '1980-guldens' salarisverhoging krijgt.

a. Dus in 1985 verdient hij 150 '1980 guldens' per avond. Dit bedrag moet nog voor inflatie gecorrigeerd worden.

Welk bedrag krijgt André per avond in 1985 werkelijk in handen?

\*) Comal:

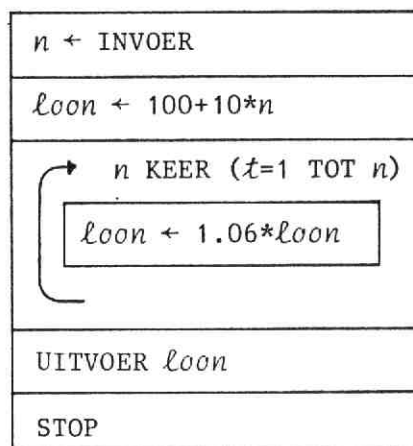
```

Regel 50 vervangen door: 50 IF T > G THEN
                        60   V := T - G
                        70 ELSE
                        80   V := G - T
                        90 ENDIF

```

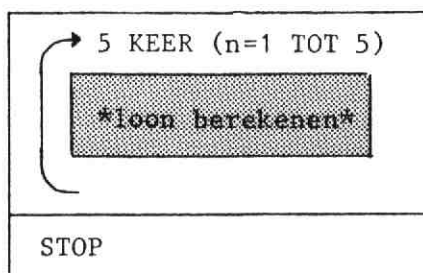
De overige regels vanaf 100 doornummeren.

- b. Ga na dat in het volgende structuurdiagram een algoritme is beschreven om André's loon voor  $n$  jaar na 1980 te berekenen.

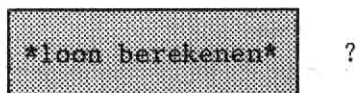


Let op: niet  $n$  maar  $t$  is de tellende variabele!

- c. Zet het structuurdiagram om in een programma en verwerk het op de computer. Neem als invoer voor  $n$  achtereenvolgens 1, 2, 3, 4, 5.
- » 32. a. Om opgave » 31c te kunnen beantwoorden heb je het programma vijf keer moeten verwerken. Erg handig is dit niet en het kan dan ook korter:



Met welk deel van het structuurdiagram van opgave » 31b correspondeert het blok:



- b. Wijzig het programma van opgave » 31c volgens het nieuwe structuurdiagram en verwerk het nog eens.

De opdracht  $R \leftarrow \text{RANDOM}$  wordt naar BASIC vertaald met  $R=\text{RND}$  \*)

» 33. a. Voorspel wat het volgende programma doet:

```
10 FOR I=1 TO 50
20 R=RND
30 G=INT(6*R+1)
40 PRINT G,
50 NEXT I
60 END
```

b. Verwerk dit programma enkele malen op de computer.

Krijg je steeds een ander resultaat? Zo nee, voeg dan aan het begin van het programma de volgende regel toe:

```
5 RANDOM      (Soms ook RANDOMIZE)
```

en verwerk het programma nog eens.

» 34. Verander het programma zo, dat het geschikt is om 50 vierkeuzevragen op de gok te beantwoorden.

» 35. Maak een computerprogramma bij opgave » 19.

» 36. Zet het structuurdiagram van opgave » 20 om in een computerprogramma en verwerk het.

\*) In plaats van  $R=\text{RND}$  soms ook  $R=\text{RND}(0)$  of iets dergelijks.  
Op sommige computers kan met  $G=\text{RND}(6)$  het gooien van een dobbelsteen rechtstreeks gesimuleerd worden.

## TRAMLIJN

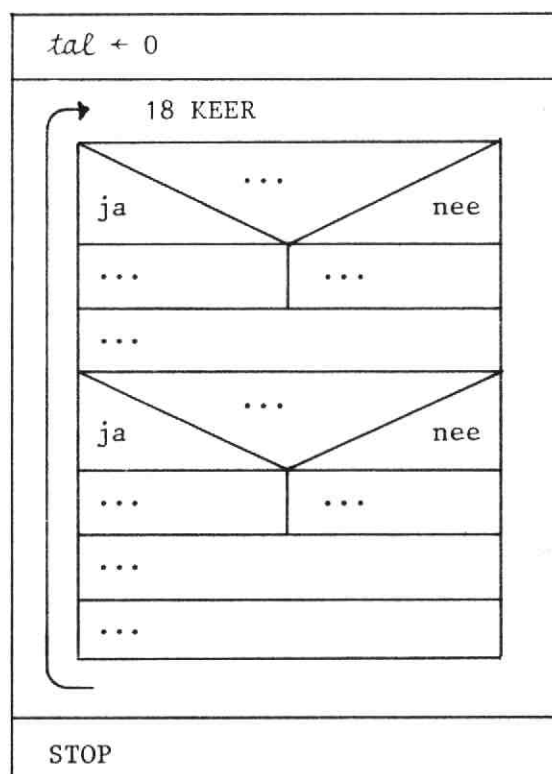
- » 37. Een tramlijn heeft 18 haltes, begin- en eindpunt meegerekend. Bij iedere halte, behalve bij het eindpunt, stappen er ten hoogste 20 mensen in. Vanaf de vierde halte stappen er hoogstens 22 mensen uit. Het programma beschrijft hiervan een simulatie.\*)

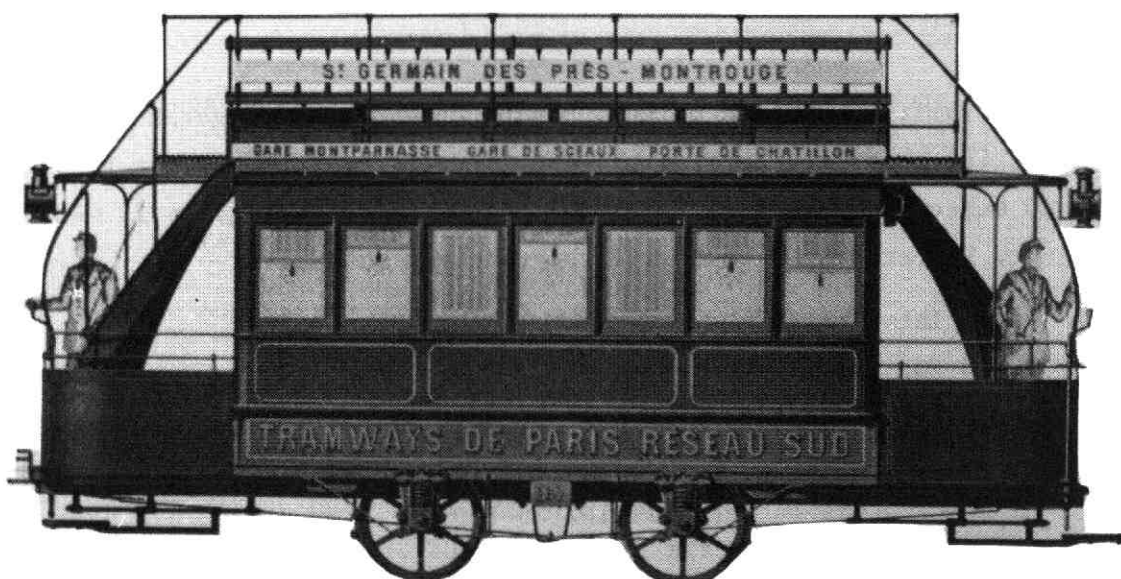
```

10 TAL=0
20 FOR HALTE=1 TO 18
30 IF HALTE<18 THEN IN=INT(21*RND(0)) ELSE IN=0
40 TAL=TAL+IN
50 IF HALTE>=4 THEN UIT=INT(23*RND(0)) ELSE UIT=0
60 TAL=TAL-UIT
70 PRINT IN, UIT, TAL
80 NEXT HALTE
90 END

```

- a. Maak het structuurdiagram dat bij dit programma hoort verder af:





*Een Parijse tram uit het eind van de 19e eeuw.*

b. Verwerk het programma.

Geef commentaar op de opgave en op de resultaten van de computerverwerking. Kun je het programma wat gebruikersvriendelijker maken?

Heb je verdere suggesties om het programma te verbeteren?

\*) Comal:

```
Het programma van opgave 37 luidt:
10 tal:=0
20 FOR halte:=1 TO 18 DO
30   IF halte <18 THEN
40     instap:=INT(21*RND)
50     tal:=tal+instap
60   ELSE
70     instap:=0
80   ENDIF
90   IF halte >=4 THEN
100    uit:=INT(23*RND)
110    tal:=tal-uit
120   ELSE
130    uit:=0
140   ENDIF
150   PRINT instap, uit, tal
160 NEXT halte
170 END
```

## OVERZICHT BASIC

In dit hoofdstuk heb je kennisgemaakt met de volgende opdrachten om een structuurdiagram om te zetten in een BASIC-programma:

*Basic-opdracht**Structuurdiagram*

INPUT A

 $a \leftarrow \text{INVOER}$ 

PRINT A

UITVOER  $a$ 

PRINT "tekst"

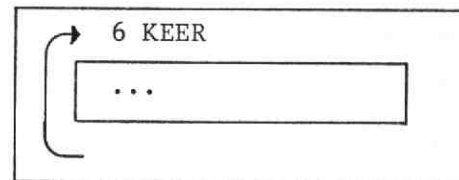
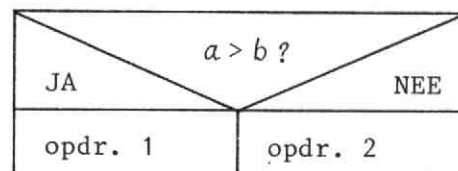
END

STOP

FOR T = 1 TO 6

...

NEXT T

IF  $A > B$  THEN opdr. 1 ELSE opdr. 2

A = RND

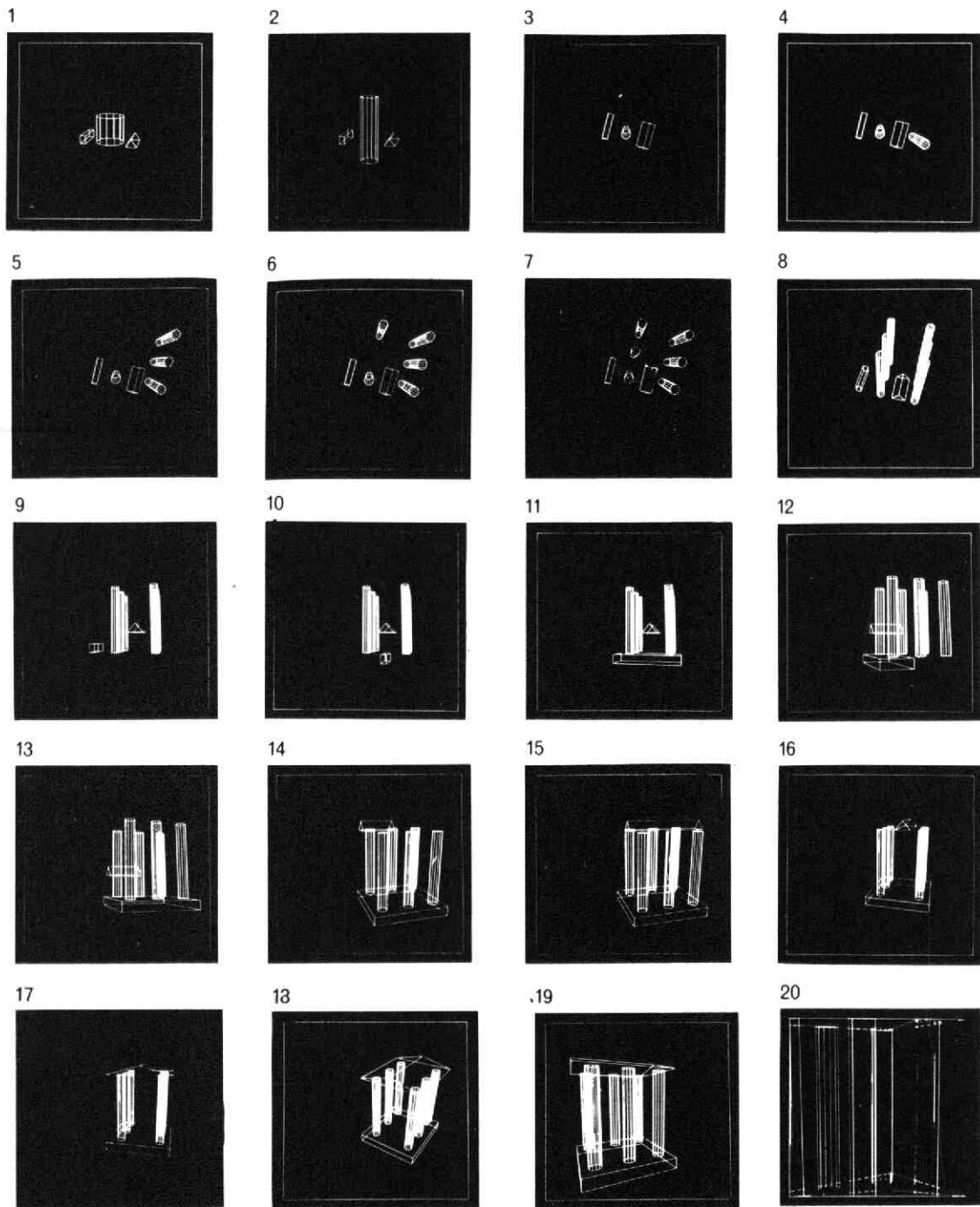
 $a \leftarrow \text{RANDOM}$ 

A = INT(B)

 $a \leftarrow \text{INT}(b)$ 

Verdere BASIC-commando's zijn:

- LIST - het programma wordt op het scherm afgedrukt
- RUN - het programma wordt verwerkt
- NEW - het programma wordt verwijderd.



### Griekse tempel

Zelfs een Griekse tempel laat zich met behulp van een computer ontwerpen! Eerst vraagt de ontwerper aan de computer om een blok, een cilinder en een driehoekig prisma uit het geheugen op te roepen. (1). Vervolgens kan de ontwerper elk van deze lichamen door de computer laten kopiëren, verschuiven, draaien en vervormen. De ontwerper verlengt de cilinder (2) en draait de drie elementen tot hij er van boven tegenaan kijkt (3). Er worden nog vijf zuilen toegevoegd (4,5,6,7). Het geheel wordt gedraaid (8) totdat de ontwerper er van voren tegenaan kijkt (9). Het blok wordt onder de zuilen gebracht (10) en in de breedte opgerekt (11). De tempel in wording wordt gedraaid tot de ontwerper er van opzij tegenaan kijkt (12) en het blok wordt in de lengte opgerekt tot de vloer gevormd is (13). Het prisma wordt opgetild (14) en verlengd (15). De tempel wordt weer gedraaid (16) en het prisma wordt verbreed tot het dak gevormd is (17). De tempel is nu klaar en wordt van boven bekeken (18). Met beide benen terug op de grond (19) loopt de ontwerper zijn tempel binnen (20).

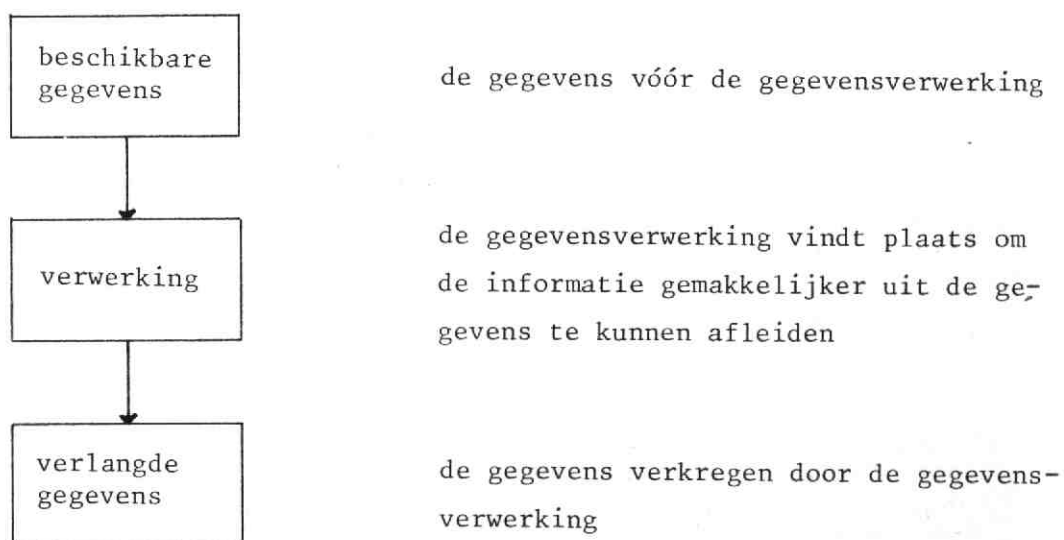
## HET WERKEN MET STANDAARDPROGRAMMA'S.

In de praktijk zijn computerprogramma's al gauw veel ingewikkelder en langer dan de programma's die je zelf in dit hoofdstuk gemaakt hebt. Voor verschillende toepassingen uit wiskunde A is het niet nodig de programma's zelf te maken, er zijn *standaardprogramma's* voor beschikbaar.

Bij het gebruik van zo'n standaardprogramma hoef je alleen de getallen voor INVOER op te geven (de *beschikbare gegevens*).

De computer *verwerkt* deze gegevens en neemt je daarmee rekenwerk uit handen. Omdat deze verwerking met behulp van een automaat plaatsvindt, spreken we van *Automatische Gegevens Verwerking*.

De resultaten van de verwerking (de *verlangde gegevens*) kun je vervolgens gebruiken om er informatie uit te halen en om er conclusies uit te trekken. Schematisch is de gang van zaken als volgt:



Het is belangrijk dat je, voordat je gaat werken met een standaardprogramma, goed weet wat je daarmee wilt.

Dus: - wat wil je precies onderzoeken?

- welke gegevens heb je tot je beschikking?

- welke informatie denk je te kunnen halen uit de computeruitvoer?

Binnen wiskunde A zal de computer onder andere gebruikt worden bij de onderwerpen Matrices, Grafische Verwerking, Kansverdelingen en Differentiëren.



