

Rijden onder invloed



De BAC-formule

Wiskunde
Biologie
Informatica

Roger Labie - Jos Punie - Eric Bogaerts - Hans Bekaert
Koen Stulens - Prof. dr. Patrick Reygel - Kurt Braeken

Opstellen van de BAC-formule (Bloed Alcohol Concentratie)

Doelstelling

Widmark heeft een formule opgesteld om op basis van een aantal parameters de bloedalcoholconcentratie (BAC) te voorspellen.

Na deze module zul je in staat zijn om:

- de formule van Widmark te interpreteren en inzicht te hebben in het gebruik van de verschillende eenheden;
- de invloed van alle optredende elementen te begrijpen;
- de verschillende reële functies te herkennen als je één parameter laat veranderen, terwijl de andere constant blijven;
- deze functies grafisch voor te stellen.

Studiemateriaal

Bij deze module hoort een werktekst waarin de formule systematisch wordt opgebouwd. Aansluitend worden de opdrachten geformuleerd.

Verloop van de module

Elke leerling neemt de werktekst zelfstandig door. Controleer alle berekeningen met je TI-83/84 Plus.

Maak alle opdrachten en controleer je antwoorden op het oplossingenblad.

De BAC-formule

De formule

De formule van Widmark wordt gegeven door

$$\text{BAC} = y = \frac{n \cdot V \cdot p \cdot 8}{r \cdot m \cdot \frac{1}{1,055}} - t \cdot \beta \quad \frac{\# \text{ gram zuivere alcohol}}{\# \text{ liter bloed}} - \text{afbraak}$$

In de teller zien we

n: aantal gedronken glazen bier, wijn, alcoholpops....

V: volume per glas in cl

p: alcoholpercentage

1 cl alcohol = 8 g

In de noemer zien we

m: lichaamsmassa in kg

r-factor of Widmarkfactor (0,68 voor mannen en 0,55 voor vrouwen)

$$1 \text{ kg bloed} = \frac{1}{1,055} \text{ l bloed}$$

De afbraak van alcohol wordt aangegeven door:

t: tijd in uren

β - factor of verteringsfactor met gemiddelde waarde $0,17 \frac{\text{g}}{\text{l.h}} = 0,17$ promille per uur

Totale lichaamsmassa en gereduceerde lichaamsmassa

Lichaamsmassa bestaat uit spieren, organen, bloed, bot, vet... Bot en vet absorberen geen alcohol.

Er zal dus slechts een deel van de lichaamsmassa alcohol absorberen. De totale massa wordt daarom met een reductiefactor vermenigvuldigd. (r-factor) Deze factor is individueel bepaald.

Meestal wordt de gemiddelde waarde gebruikt: voor mannen is dit 0,68 en voor vrouwen 0,55.

Vertering van alcohol in ons lichaam

Enzymen zorgen voor vertering van alcohol. Deze verteringsfactor is ook persoonsgebonden. Meestal nemen we de gemiddelde waarde van $0,17 \text{ g/l.h}$.

In België is de toegestane BAC-waarde = $0,5 \frac{\text{g}}{\text{l}} = 0,5$ promille

Parameters die de BAC beïnvloeden

Lichaamsmassa als parameter

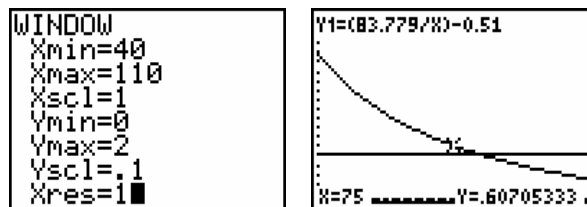
Een man drinkt 6 glazen bier van 25 cl op 3 uur tijd. Het alcoholgehalte is 4,5%.

$$\begin{aligned} y = \text{BAC} &= \frac{(6 \cdot 25 \cdot 0,045) \text{ cl} \cdot 8 \frac{\text{g}}{\text{cl}}}{(0,68 \cdot x) \text{ kg} \cdot \frac{1}{1,055} \frac{\text{l}}{\text{kg}}} - 3 \text{ h} \cdot 0,17 \frac{\text{g}}{\text{l.h}} \\ &= \frac{83,7794}{x} - 0,51 \frac{\text{g}}{\text{l}} \end{aligned}$$

Deze hyperbolische functie kan getekend worden door een verticale uitrekking gevolgd door een verticale verschuiving van $y = \frac{1}{x}$. De veranderlijke x is de lichaamsmassa in kg.

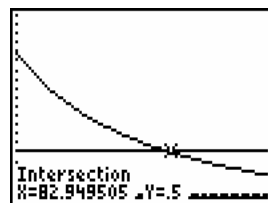
Volg deze werkwijze:

- Voer in je TI-83/84 Plus de functie in en ook $y = 0,5$ (toegestane BAC waarde).
- Met $\boxed{2nd}$ CALC en 1: VALUE kan je een willekeurige lichaamsmassa invoeren en de BAC waarde berekenen.



Voor iemand van 75 kg bekom je BAC = 0,607...

- Met $\boxed{2nd}$ CALC en 5: INTERSECT bepaalt men het snijpunt.



De persoon moet ten minste 83 kg wegen om met de wagen te mogen rijden.

- Met $\boxed{2nd}$ TBLSET (TblStart = 40, Δ Tbl = 1) en $\boxed{2nd}$ TABLE kan men de BAC aflezen vanaf 40 kg.

TABLE SETUP			
TblStart=40			
Δ Tbl=1			
Indent: Auto	Ask		
Depend: Auto	Ask		

X	Y1	Y2
77	.57804	.5
78	.56409	.5
79	.55049	.5
80	.53724	.5
81	.52431	.5
82	.5117	.5
83	.49939	.5

Voor dezelfde gegevens maar met een r-factor van 0,55 (vrouw) bekommen we nu de functie:

$$BAC = y = \frac{103,582}{x} - 0,51$$

Dit betekent dat voor een vrouw van 70 kg de BAC-waarde 0,97..‰ is. Je kunt nagaan dat een vrouw niet meer strafbaar is vanaf 102,5 kg.

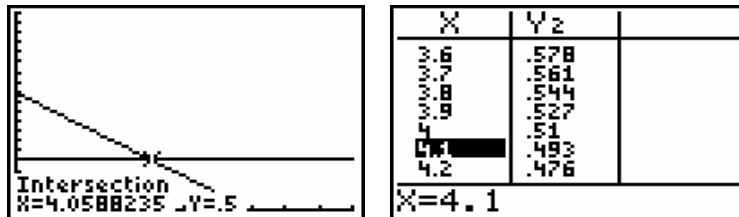
Tijd als parameter

Een man van 70 kg drinkt 6 glazen bier met 4,5% alcoholgehalte. Na hoeveel uur is hij niet meer strafbaar?

$$y = \text{BAC} = \frac{(6.25 \cdot 0,045) \text{ cl} \cdot \frac{\text{g}}{\text{cl}}}{(0,68.70) \text{ kg} \cdot \frac{1}{1,055} \frac{\text{l}}{\text{kg}}} - 0,17 \frac{\text{g}}{\text{l} \cdot \text{h}} \cdot x$$

$$= 1,19... - 0,17 \frac{\text{g}}{\text{l} \cdot \text{h}} \cdot x$$

Welke functie is dit? Is deze functie stijgend of dalend?

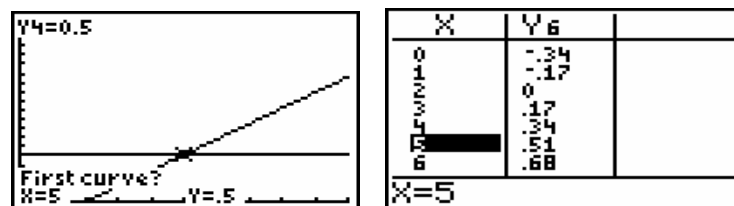


Aantal drankjes als parameter

Een man van 80 kg drinkt gespreid over twee uur een aantal pintjes bier op café. Hoeveel pintjes mag hij drinken om niet strafbaar te zijn?

$$y = \text{BAC} = \frac{(25 \cdot 0,045) \text{ cl} \cdot \frac{\text{g}}{\text{cl}}}{(0,68.80) \text{ kg} \cdot \frac{1}{1,055} \frac{\text{l}}{\text{kg}}} \cdot x - 2 \text{ h} \cdot 0,17 \frac{\text{g}}{\text{l} \cdot \text{h}}$$

$$= 0,17... \cdot x - 0,34 \frac{\text{g}}{\text{l}}$$



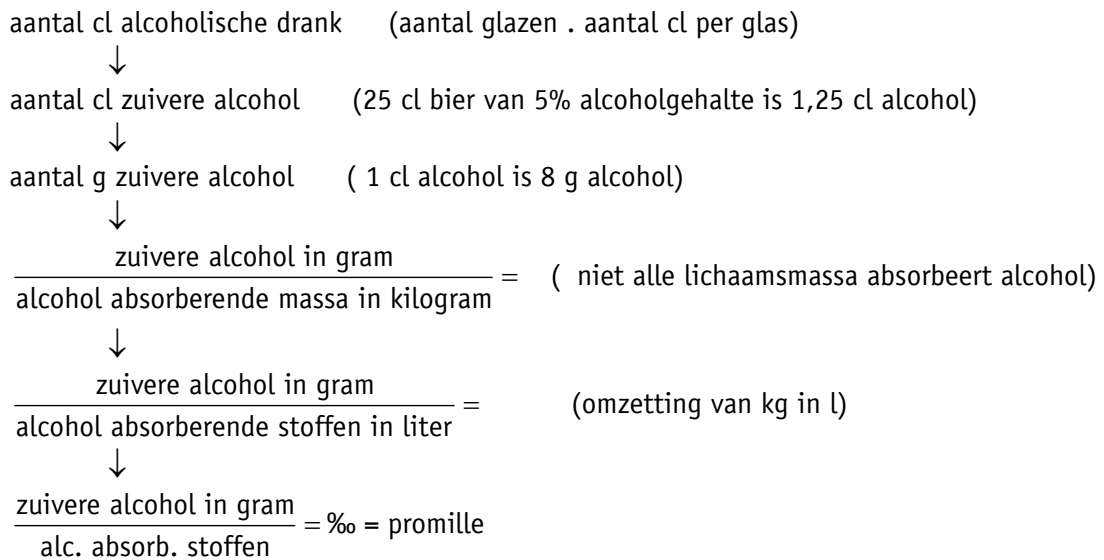
Opdrachten

Opdracht 1

De eenheid van BAC is $\frac{\text{mg}}{\text{ml}} = \frac{\text{g}}{\text{l}} = \text{‰}$ en wordt promille genoemd.

Hoe kom je uiteindelijk tot deze eenheid?

schema:



Opdracht 2

Een man van 70 kg drinkt 6 pilsjes van 25 cl en 4,5% alcoholpercentage op 3 uur tijd. Zijn r-factor is 0,68 en β -factor is $0,17 \frac{\text{g}}{\text{L.h}}$. Bereken zijn BAC.

Opdracht 3

Twee meisjes van 60 kg hebben een respectievelijke r- factor van 0,6 en 0,5. Wat betekent dit voor hun bot en vetmassa? Wie van hen heeft de grootste BAC als alle andere factoren gelijk zijn? Dit moet je wiskundig duiden.

Vul in: Hoe meer bot- en vetmassa, hoe.....alcoholabsorberende massa, hoe.....de r- factor, hoe de BAC.

Opdracht 4

We spreken af om voortaan de gemiddelde r-factor 0,68 voor mannen en 0,55 voor vrouwen en de gemiddelde β -factor $0,17 \frac{\text{g}}{\text{L.h}}$ te nemen. De hoeveelheid alcohol, de lichaamsmassa en het aantal uren van de vertering blijven dan de onafhankelijke variabelen van de BAC.

- neem het aantal gram alcohol als variabele
- neem het aantal uren als variabele
- neem de lichaamsmassa als variabele

Geef een algemene vergelijking van deze functies.

Wat is het essentiële verschil bij de functies in a en b? Maak een schets om het onderscheid duidelijk te maken. (tip: denk aan het teken van de coëfficiënten)

Hoe teken je de functie van c vertrekkende van $y = \frac{1}{x}$.

Opdracht 5

Een student drinkt 7 pilsjes van 25 cl en 4,5% alcoholgehalte op 2 uur tijd. Stel de functie op en plot ze met je TI-83/84 Plus. (Kies goede vensterinstellingen). Welke lichaamsmassa moet hij hebben om niet strafbaar te zijn? (oplossing bepalen met je TI-83/84 Plus)

Opdracht 6

Een vrouw van 62 kg drinkt 7 glazen wijn van 12 cl en 13% alcoholgehalte. (denk aan de gemiddelde r-factor en verteringsfactor) Hoe lang moet zij op café blijven om ongestraft met de wagen te rijden?

Opdracht 7

Verklaar waarom twee mannen van respectievelijk 70 en 75 kg die evenveel alcohol drinken over dezelfde tijd en met dezelfde verteringsfactor toch dezelfde BAC kunnen hebben?

Opdracht 8

Twee vrienden Jan en Bart wegen 70 kg en drinken op 4 uur tijd een aantal softdrinks van 25 cl en 5% alcoholpercentage. De frêle maar zeer gespierde Bart heeft een r-factor 0,75, terwijl Jan corpulent is en een r-factor van 0,62 heeft. Ze hebben beide een gemiddelde verteringsfactor $0,17 \frac{\text{g}}{\text{L.h}}$.

Hoeveel glazen mogen ze drinken om met de auto nog naar huis te kunnen rijden?

Opstellen van de formule (W)

Verklaring van de formule (W+B)

Programmeren van de formule (I)

Verklaring van de BAC-formule (Bloed Alcohol Concentratie)

Biologische factoren bij het berekenen van BAC met behulp van de Widmark-formule.

De formule van Widmark wordt gegeven door

$$BAC = \frac{n \cdot V \cdot p \cdot 8}{r \cdot m \cdot \frac{1}{1,055}} - t \cdot \beta$$

met n het aantal glazen alcohol,
 V het volume van één glas in cl,
 p het percentage alcohol
 r de Widmark-factor
 m de lichaamsmassa in kilogram
 t de tijd in uren
 β de afbraaksnelheid in g/l.h

Om beter inzicht te krijgen in deze formule willen wij hieronder de biologische achtergronden van enkele parameters verduidelijken.

(A) De Widmark factor r

De Widmark factor r is een correctiefactor die op basis van de lichaamsmassa de hoeveelheid bloedmassa bepaalt, die alcohol absorbeert. Deze factor is bij mannen anders dan bij vrouwen.

www.dui-law.com/810art.htm	www.nhtsa.dot.gov/people/injury/alcohol/bacreport.html	Boek Human pharmacology
Man: 0,68 (range van 0,5 tot 0,9 met gemiddelde 0,68) Vrouw: 0,55 (range van 0,5 tot 0,9 met gemiddelde 0,68)	0,58 (geen rekening gehouden met man of vrouw)	Man: 0,68 Vrouw: 0,55

De gegevens uit kolom 1 en 3 bevestigen de waarden die gebruikt zijn in de vorige paragraaf.

Deze correctiefactor is te wijten aan het feit dat botten en vet nl. geen alcohol op nemen. Bij mannen is de som van deze twee iets lager dan bij vrouwen. Het totaal van deze twee fracties moet dus afgetrokken worden van de totale lichaamsmassa.

In de onderstaande tabel (<http://www.wvc.vlaanderen.be/gezondsporten/index.htm>), vinden we een bevestiging van deze correctiefactoren. Deze tabel illustreert cijfermatig de lichaamssamenstelling van de “referentiemann” en –“vrouw”, in termen van spier, vet en bot.

	Referentieman	Referentievrouw
Leeftijd	20 – 24 j	20 – 24 j
Lich. lengte	174,0 cm	163,8 cm
Lichaamsmassa	70 kg	56,7 kg
Totaal vet	10,5 kg (15,0 %)	15,3 kg (27,0 %)
Depot vet	8,4 kg (12,0 %)	8,5 kg (15,0 %)
Essentieel vet	2,2 kg (3,0 %)	6,8 kg (12,0 %)
Spier	31,3 kg (44,8 %)	20,4 kg (36,0 %)
Bot	10,4 kg (14,9 %)	6,8 kg (12,0 %)
Residu	17,6 kg (25,3 %)	14,1 kg (25,0 %)
Lean body mass	61,7 kg	48,5 kg
a. essentieel vet	1,86 kg (3,0 %)	6,8 kg (14,0 %)
b. spier	30,8 kg (50 %)	20,4 kg (42,0 %)
c. bot	10,4 kg (17,0 %)	6,8 kg (14,0 %)

Tel je vet en bot op voor mannen dan kom je aan 29,9 % (rest dus 70% = 0,7.)

Voor vrouwen is dat samen 39% (rest dus 60% = 0,6.)

Het gaat hier wel om een beperkte leeftijdsgroep (20-24j) wat dus de kleine afwijking t.o.v. de Widmark r-factor kan verklaren.

(B) De soortelijke massa van bloedplasma

De soortelijke massa van bloedplasma is hoger dan die van water en is gelijk aan 1,055 g/ml. Dat komt door allerlei stoffen die in het bloed opgelost zijn: nutriënten zoals aminozuren en sachariden, eiwitten zoals fibrinogeen en allerlei mineralen.

Om het juiste bloedvolume in liter te bekomen moeten wij dus de bloedmassa delen door de soortelijke massa van het bloedplasma.

(C) Afbraaksnelheid (Widmark β -factor)

De afbraaksnelheid bepaalt in welke mate het lichaam alcohol afbreekt. Deze waarde is afhankelijk van het feit of je al dan niet een ervaren drinker bent en hoe de enzymen invloed hebben op het afbraakproces. Omwille van vereenvoudiging nemen we een gemiddelde β -waarde.

Een afbraaksnelheid van 0,17 g/(l.h) (= 0,17 promille/h) komt overeen met het feit dat je iets meer dan een uur nodig hebt om een standaardglas alcohol af te breken.

Deze waarde zal in principe afhangen van de lichaamsmassa en het geslacht van de persoon. De afbraaksnelheid kan enorm verschillen van persoon tot persoon (volgen Human pharmacology met een verhouding van 0,05 tot 30 mmol/(l.h) = een verhouding van 600!). Bij regelmatige drinkers komen er ook bijkomende processen op gang in de lever, die de afbraaksnelheid verhogen.

Bij het gebruik van de Widmark-formule moet alleszins beklemtoond worden dat geen enkele formule in staat is de individuele BAC feilloos te bepalen. De reductiefactor en afbraakfactor zijn individueel, terwijl in de formule gemiddelde waarden worden gebruikt.

Programmeren met de TI-83/84 Plus

Inleiding

Wanneer je berekeningen die zich steeds herhalen eigenlijk liever door een computer zou willen laten uitvoeren, dan kun je best zelf leren hoe eenvoudige computerprogramma's kunnen worden gemaakt. Daardoor moet je ingewikkelde formules die je maar af en toe gebruikt of vaak aanleiding geven tot fouten niet meer zelf onthouden. Deze tekst handelt over de manier waarop een programma moet worden opgebouwd met behulp van de TI-83/84 Plus.

Doelstelling

Deze module wil je begeleiden in het zelf leren programmeren.

Het einddoel is een programma schrijven op de grafische rekenmachine waarmee je kunt nagaan of je in staat bent om een wagen te besturen, op basis van je lichaamskenmerken en het aantal gedronken glazen bier of wijn.

Na het doornemen van deze module zal je in staat zijn

- programmacode van bestaande programma's te begrijpen en aan te passen;
- zelf programma's te schrijven in de programmeertaal TI-Basic;
- de formule van Widmark op een geautomatiseerde manier gebruiken zodat je snel kunt weten of je te veel alcohol gedronken hebt of niet

Studiemateriaal

De leerstof voor deze module wordt je aangeboden in de vorm van een werktekst. De tekst start met een algemene inleiding en evolueert snel naar concrete voorbeelden en toepassingen. Theoretische beschouwingen en oefeningen worden afgewisseld. Na elk onderdeelje met nieuwe theorie wordt een aantal oefeningen gegeven om de aangebrachte leerstof in te oefenen.

Zorg dat je ook je handleiding van de TI-83/84 Plus bij de hand hebt. Het is handig om wat meer achtergrondinformatie ter beschikking te hebben. Eventueel kun je de handleiding (guidebook) downloaden van <http://education.ti.com>.



"Kreeg je mijn mailtje niet?"

Afspraken

Ga heel systematisch te werk: lees de tekst aandachtig, noteer je programma's eerst in een schrift of map, ga dan pas over tot het ingeven van de code in het rekentoestel. Zorg dat je op elk ogenblik je code kunt voorleggen zodat foutjes snel opgespoord kunnen worden.

Bij deze module zijn ook hints ter beschikking als hulp bij het opstellen van je programmacode. Gebruik deze hints enkel nadat je zelf eerst gezocht hebt naar een oplossing van je probleem.

Opdrachten

Wat is programmeren?

Programmeren kunnen we omschrijven als het opbouwen van algoritmen. Wanneer we algoritmen gaan schrijven die door een computer moeten worden uitgevoerd, spreken we over programma's.

Wat is een algoritme?

Een **algoritme** is een voorschrift om in een eindig aantal stappen een taak uit te voeren. Het is dus een handelingsvoorschrift. Het is een opsomming van alle handelingen die nodig zijn om een bepaald probleem op te lossen. Handelingen worden opgedeeld in deelhandelingen, die ook weer kunnen worden opgedeeld in kleinere stapjes...

Voorbeelden van algoritmen:

- Breipatroon
- Montagehandleiding bij een IKEA-kast
- Recept in een kookboek...

Wat is een programma?

Een **programma** is een opeenvolging van een aantal instructies die kunnen worden uitgevoerd door een verwerkingseenheid (processor). Het is een algoritme, uitgewerkt in een programmeertaal.

Voorbeelden van programma's:

- Microsoft® Windows
- TI InterActive!
- Een programma om snel de discriminant van een tweedegraadsterm te bepalen.

Verschil tussen een algoritme en een programma:

Een programma is een algoritme dat neergeschreven is in een programmeertaal. Een algoritme is onafhankelijk en kan dus in elke willekeurige programmeertaal worden uitgeschreven.

Elk programma wordt steeds opgebouwd vanuit een degelijk doordacht algoritme (niet andersom).

Wat is een programmeertaal?

Programmeren gebeurt in een programmeertaal. Er bestaan tientallen programmeertalen:

- Basic
- Pascal
- Fortran
- C++
- Java
- ...

Een beetje geschiedenis

De eerste generatie programmeertalen stamt uit de periode 1950. Men sprak toen over een machinetaal die rechtstreeks door de processor werd uitgevoerd. De taal was dus afhankelijk van de processor die het programma moest uitvoeren.

De tweede generatie programmeertalen noemt men assembleren (1955). Deze taal werd ontwikkeld per processorfamilie en wordt vandaag nog steeds gebruikt.

De derde generatie programmeertalen zijn de hogere programmeertalen (1960). De programmeercode was leesbaar in het Engels. De belangrijkste voorbeelden zijn FORTRAN (Formula Translation - 1957) voor wetenschappelijke toepassingen, COBOL (Common Business Oriented Language - 1960) voor administratieve toepassingen, BASIC (Beginners All-purpose Symbolic Instruction Code - 1964), speciaal ontwikkeld voor computerleken en PASCAL (naar Blaise Pascal - 1971), dat vooral een didactische taal was.

De 4e generatie taal is ontwikkeld rond 1980. Het was een taal om gegevensbanken te ondervragen (vb. SQL)

De 5e generatie taal stamt uit de periode 1990 en was bedoeld voor kunstmatige intelligentie & neurale netwerken. De taal van de toekomst is Java (ontwikkeld door SUN microsystems - 1994).

Welke programmeertaal?

De keuze van een taal hangt af van de machine die het programma moet uitvoeren. Al de computertalen steunen op dezelfde basisprincipes, die we gemakkelijk kunnen aanleren via BASIC. De taal die we hier gebruiken is een variant van deze taal en heet TI-BASIC. Vanuit de kennis van deze taal, kun je gemakkelijk overstappen naar andere programmeertalen zoals FORTRAN of PASCAL.

De creatie van een programma

Het schrijven van een programma gebeurt in verschillende stappen:

1. Omschrijf precies wat het gestelde probleem inhoudt.
2. Analyseer het probleem en zoek naar de beste oplossing.
3. Stel een algoritme op met de oplossing van het probleem. Werk de oplossing stap voor stap uit.
4. Vertaal het algoritme naar de broncode. De gebruikte code moet volledig correct zijn. Men noemt dit het respecteren van de syntax. De broncode wordt nadien vertaald naar een uitvoerbaar bestand. Dit noemt men compileren. Het resultaat is een programma dat door de processor kan worden verwerkt.
5. Test je programma.

Om een goed programma te schrijven, moet je vertrouwd zijn met het probleemgebied waar het programma betrekking op heeft en moet je bijgevolg in staat zijn om het probleem te vertalen naar een algoritme.

Voorbeeld:

Je kunt je afvragen hoe je de Body-Mass Index van een persoon precies moet berekenen:

1. Vraag de massa van de persoon, uitgedrukt in kilogram (kg).
2. Vraag de lengte van de persoon, uitgedrukt in meter (m).
3. Deel de massa van de persoon door het kwadraat van de lengte.
4. De BMI-waarde is het bekomen getal.

Het algoritme bestaat hier uit vier stappen. Deze vier stappen zul je ook terugvinden in het programma dat je wilt schrijven om de BMI-waarde te berekenen.

De onderdelen van een programma

Globaal genomen bestaat een programma uit drie onderdelen:

1. INVOER
2. VERWERKING
3. UITVOER

Je programma heeft een aantal gegevens nodig om de berekening te kunnen starten (**invoer**). In het voorbeeld van de berekening van de BMI-waarde is dat de massa en de lengte van de persoon.

De **verwerking** van de gegevens is de kern en het hoofddoel van het programma: vb. een snelle en correcte berekening van de BMI-waarde.

Bij de **uitvoer** wordt het resultaat van de verwerking getoond.

De syntax van een programma

Elk programma is opgebouwd volgens vaste regels die strikt moeten gevolgd worden. Anders wordt het programma niet begrepen door de machine die het moet uitvoeren. Het programma zal vastlopen en geen resultaat of een foutmelding geven. Elke programmeertaal heeft z'n eigen commando's. Een aantal belangrijke commando's in TI-BASIC zijn bijvoorbeeld:

DISP " ..."	Toont de tekst tussen de aanhalingstekens ¹
CLRHOME	Wist het scherm van de grafische rekenmachine (GRM)

In een programma worden variabelen gebruikt. Getallen die worden ingevoerd krijgen een plaats in het geheugen die wordt aangeduid met één letter. De inhoud van die geheugenplaats noemen we de **variabele**. De gebruikte letter is de **naam van de variabele**. Naar de inhoud wordt telkens verwezen, door gebruik te maken van die naam.

Variabelen hoeven niet alleen cijfers te zijn. Een variabele kan ook een functievoorschrift zijn, een stukje tekst, een tekening, ... De bewerkingen die je op deze variabele toepast, moeten uiteraard gedefinieerd zijn voor de betrokken variabele. Indien je een verkeerde bewerking op een variabele probeert uit te voeren, zal het programma vast lopen en een foutmelding geven.

Een aantal voorbeelden van commando's die werken met variabelen:

PROMPT A	Vraagt naar de inhoud van de variabele A
DISP A	Toont de inhoud van de variabele A
DISP "DE INHOUD VAN A IS ", A	Toont de zin tussen de aanhalingstekens, gevolgd door de inhoud van A.
2A→B	Stopt de dubbele waarde van A in de variabele B (de pijl wordt ingegeven met de  -toets)

Voorbeeld: het berekenen van de BMI-waarde. Ga na of je deze code goed begrijpt. Wacht nog even met het intikken van de code.

```
:DISP "GEEF JE MASSA"  
:PROMPT M  
:DISP "GEEF JE LENGTE"  
:PROMPT L  
:M/L2→B  
:DISP "JE BMI-WAARDE IS ",B
```

¹ Om tekst in te geven, moet je rekentoestel in de ALPHA-mode staan. Tussen de aanhalingstekens kun je maximaal 16 tekens plaatsen. Het dubbele accent-teken vind je in de ALPHA-mode bij de -toets.

Merk op dat de eerste en de derde regel niet echt noodzakelijk zijn om je programma te laten werken. Ze maken het programma wel gebruiksvriendelijker en beter leesbaar. Dat zul je merken als je het programma intikt en uitvoert.

Programma-menu van de TI-83/84 Plus

Indien je de **[PRGM]**-knop activeert vanuit het hoofdscherm, dan krijg je de keuze uit drie menu's:

- **[EXEC]**: om programma's die je reeds eerder hebt gemaakt uit te voeren (execute)
- **[EDIT]**: om programma's die je reeds eerder hebt gemaakt aan te passen (edit)
- **[NEW]**: om een nieuw programma te beginnen.

```
EXEC EDIT NEW
```

```
1: I/O EXEC
2: If
3: Then
4: Else
5: For(
6: While
7: Repeat
8: End
```

Indien je **[PRGM]**-knop activeert terwijl je programmacode aan het ingeven of verbeteren bent, krijg je eveneens de keuze uit drie menu's:

- **[CTL]**: om controlestructuren te maken (control)
- **[I/O]**: om in- of uitvoer op te vragen of te tonen (input/output)
- **[EXEC]**: om een andere programma op te roepen (execute)

Oefening 1:

Voer de code van het BMI-programma nu in in je grafische rekenmachine. Je geeft je programma eerst een naam en daarna geef je op elke regel één commando in. Een nieuwe regel begint telkens met een dubbele punt. Het rekentoestel plaatst dit automatisch.

```
EXEC EDIT NEW
1: Create New
```

1. Om je programma te maken, druk je **[PRGM]** > **[NEW]** > 1: CREATE NEW **[ENTER]**
2. Je geeft een naam² in (vb. BMI) en bevestigt met **[ENTER]**.
3. Je kunt starten met het ingeven van de programmacode. Overloop misschien eerst de verschillende commando's in elk menu door op de **[PRGM]**-toets te drukken. Druk eventueel **[CLEAR]** om naar het invoerscherm terug te keren.
4. Tik **[ENTER]** om een nieuwe lijn te beginnen.
5. Om te eindigen met het invoeren van je code, druk je **[2nd]** **[QUIT]**³
6. Om je programma te starten, druk je **[PRGM]** > **[EXEC]** > 1: BMI

```
PROGRAM
Name=BMI
```

```
PROGRAM: BMI
:
```



Tijdens het invoeren kun je een regel wissen door op **[CLEAR]** te drukken. Een regel tussenvoegen doe je door bij het begin van de regel te gaan staan en op **[2nd]** **[INS]** **[ENTER]** te drukken. Een regel verwijderen doe je met **[DEL]**.

De uitvoer functie DISP heeft als nadeel dat er twee regels voor gebruikt worden. Met behulp van de functie `OUTPUT(regel, positie, waarde)` kun je een mooier resultaat bereiken. Voor *regel* geef je een nummer van 1 tot 8 (het scherm is acht lijnen hoog) en voor *positie* een nummer van 1 tot 16 (het scherm is 16 karakters breed). Indien de *waarde* een tekst is, plaats je die tussen aanhalingstekens. Om de regelnummers goed te laten kloppen, kun je best je programma beginnen met het wissen van het scherm via `ClrHome`.

² Een programmaam mag maximaal acht tekens bevatten en mag niet beginnen met een cijfer.

³ Een programma dient niet opgeslagen te worden. Dat gebeurt automatisch.

Het bovenstaande voorbeeldje kun je dus omvormen tot

```
:CLRHOME
:DISP "GEEF JE MASSA"
:PROMPT M
:DISP "GEEF JE LENGTE"
:PROMPT L
:M/L2→B
:OUTPUT(6,1,"JE BMI IS ")
:OUTPUT(6,10,B)
```

Test het verschil even uit. Om een programma herhaaldelijk uit te voeren, kun je na het beëindigen gewoon op **ENTER** drukken. Het programma start dan opnieuw. Een programma onderbreken, kun je via **2nd** QUIT.

Oefening 2

We schrijven nu een kort programmaatje dat voor een veelterm van de tweede graad de bijhorende discriminant berekent.

- Noteer eerst het algoritme.
- Vertaal dit algoritme naar TI-BASIC. Noteer de volledige programmacode alvorens je ze ingeeft in je reken toestel. Wanneer je het algoritme puur ingeeft als programma is het weinig gebruiksvriendelijk. Voeg daarom best nog wat tekst toe om het programma beter leesbaar te maken.
- Geef nu de programmacode in in je grafische rekenmachine.

Opmerking:

In deze oefening hebben we de verschillende stappen van het programmeren doorlopen:

1. Analyse van het probleem: het opstellen van het algoritme
2. Vertalen van het algoritme naar programmeertaal
3. Het verfijnen van de programmeertaal

Deze werkwijze noemt men een "**top-down**"-ontwikkeling.

Controlestructuren

De bovenstaande programma's kunnen nu verder uitgewerkt worden. Zo kun je de GRM laten aangeven of de BMI-waarde goed is of onrustwekkend, of je kunt de GRM laten aangeven hoeveel wortels de tweedegraadsvergelijking zal hebben.

Beide voorstellen vragen een controle van het bekomen resultaat: op basis van het resultaat wordt een verschillend stuk van het programma verder afgewerkt. Dergelijke commando's noemt men **controlestructuren**. We overlopen de verschillende mogelijkheden. We testen telkens de controlestructuren met korte voorbeelden.

De controlestructuren vind je tijdens het intikken van de programmacode bij **PRGM** > [CTL].

If, then, end

Deze controle test een voorwaarde, die geformuleerd wordt achter IF. Indien aan de voorwaarde is voldaan wordt uitgevoerd wat achter THEN staat. Indien niet aan de voorwaarde wordt voldaan, wordt verder gegaan met wat achter END staat.

De syntax van deze controle ziet er als volgt uit:

```
:If voorwaarde  
:Then  
:opdracht (indien waar)  
:opdracht (indien waar)4  
:End  
:opdracht
```

Voorbeeld

We maken een programmaatje dat vraagt naar een getal dat kleiner of gelijk is aan tien. Indien het ingetypte getal te groot is, zegt het programma dat het ingegeven getal niet voldoet. Het programma krijgt de naam TEST1.

```
:Prompt C  
:If C>10          (<, >, =,... vind je onder de knop 2nd TEST)  
:Then  
:DISP "TE GROOT"  
:End
```

```
PROGRAM:TEST1  
:Prompt C  
:If C>10  
:Then  
:Disp "TE GROOT"  
:End
```

Je kunt ook meer ingewikkelde voorwaardes formuleren, door gebruik te maken van operatoren: OR en AND, die je vindt onder 2nd TEST > [LOGIC].

Voorbeeld

If A<10 AND A≥0	Indien A kleiner is dan tien EN groter of gelijk is aan nul
If C=1 OR C=2	Indien C gelijk is aan 1 of 2

Oefening 3

Maak een programma dat vraagt naar een cijfer tussen nul en honderd. Indien het cijfer kleiner is dan nul of groter dan honderd, geeft het programma de melding "slechte invoer". Noem dit programma OEF3.

If, Then, Else, End

Deze controle test een voorwaarde, die genoteerd staat na IF. Indien aan de voorwaarde is voldaan, wordt uitgevoerd wat achter THEN staat. Indien aan de voorwaarde niet is voldaan, wordt uitgevoerd wat achter ELSE staat. De controle wordt afgesloten met END.

De syntax van deze controle:

```
:If voorwaarde  
:Then  
:opdracht (indien waar)  
:opdracht (indien waar)  
:Else  
:opdracht (indien NIET waar)  
:opdracht (indien NIET waar)  
:End
```

⁴ Het feit dat er twee keer *opdracht (indien waar)* staat is om duidelijk te maken dat je verschillende programmaregels kunt plaatsen tussen THEN en END. Al deze programmaregels worden uitgevoerd als de voorwaarde vervuld is.

:opdracht

Merk op dat je in de opdrachten na THEN of ELSE ook een nieuwe controle kunt laten uitvoeren. Dit noemt men het **nesten** van controlestructuren.

Oefening 4

Maak een programma dat vraagt naar een cijfer tussen nul en honderd. Indien het cijfer groter of gelijk is aan 50, krijg je als melding "Geslaagd!", indien het resultaat kleiner is dan 50 krijg je als melding "Niet geslaagd!". Het programma moet net zoals in oefening 3 melding maken van een verkeerde invoer. Noem dit programma OEF4.



Indien je dat wenst kun je in een nieuw programma gebruik maken van een vorig programma. Je gaat als volgt te werk:

1. Creëer een nieuw programma en geef het een nieuwe naam.
2. Druk op **[2nd]** RCL (Recall).
3. Roep nu het programma op dat je als basis wilt gebruiken via **[PRGM]** > **[EXEC]**. Kies het juiste programma uit de lijst. De programmacode verschijnt nu in het nieuwe programma met de nieuwe naam.
4. Je kunt de code nu vervolledigen of aanpassen.

Oefening 5

Maak een programma dat voor een tweedegraadsvergelijking nagaat hoe groot de discriminant is. Als uitvoer krijg je de discriminant en de bijhorende nulpunten. Indien de discriminant kleiner is dan nul, krijg je een melding "Geen oplossing". Noem dit programma VGL.

Labels

Om een sprong in een programma te maken, kun je gebruik maken van labels. LBL (label) en GOTO (ga naar) zijn instructies die samen gebruikt worden om sprongen binnen een programma uit te voeren.

Lbl bepaalt het label voor een opdracht. Een label kan bestaan uit één of twee tekens.

Goto zorgt ervoor dat het programma verder wordt uitgevoerd vanaf label wanneer de GOTO-instructie wordt ontmoet. Beide commando's vind je onder **[PRGM]** > **[CTL]**.

```
PROGRAM: MACHT
:Lbl 0
:Prompt A
:If A≥100
:Stop
:Disp A^3
:Goto 0
```

Dit programmaatje vraagt de waarde van A, indien die groter is dan 100 stopt het programma. Indien de waarde kleiner is dan 100, berekent het de derdemacht van het gegeven getal. Daarna gaat het programma verder tot GOTO. Hier springt het programma weer naar de beginregel.

Je kunt in combinatie met Lbl ook het commando Menu(gebruiken. Met deze instructie kun je binnen het programma een sprong maken op basis van een keuze uit het voorgestelde menu.

De syntax ziet er als volgt uit:

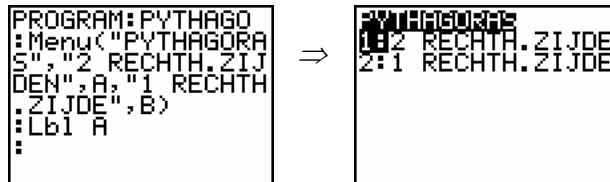
Menu("menutitel","tekst 1",label1,"tekst 2",label2,...)



De titel en de tekst komen tussen aanhalingstekens. Ook dit commando vind je onder **PRGM** > [CTL].

Oefening 6

Schrijf een programma dat bij invoer van twee zijden van een rechthoekige driehoek automatisch de derde zijde berekent. Laat de gebruiker bij het begin van het programma kiezen voor welke zijden worden opgegeven: twee rechthoekzijden of een schuine zijde en een rechthoekzijde.



Om het programma te laten stoppen na het uitvoeren van een programmaonderdeel, kun je het commando STOP gebruiken. Je voert dit bijvoorbeeld in vlak voor je een nieuw label creëert. Je vindt het in het menu **PRGM** > **[CTL]**.

Programmalussen

Je kunt tijdens een programma een stuk van de code verschillende keren na elkaar laten uitvoeren. Een dergelijk stukje code noemt men een **programmalus**. Je hebt de keuze tussen twee systemen:

1. De WHILE-lus test de voorwaarde vóór de code wordt uitgevoerd. Zolang aan de voorwaarde is voldaan, wordt de code opnieuw uitgevoerd.

```

:While voorwaarde
:opdracht (zolang voorwaarde waar is)
:opdracht (zolang voorwaarde waar is)
:End
:Opdracht

```

2. De REPEAT-lus test de voorwaarde nadat de code is uitgevoerd. De code wordt herhaald totdat aan de voorwaarde is voldaan.

```

:Repeat voorwaarde
:opdracht (totdat voorwaarde waar is)
:opdracht (totdat voorwaarde waar is)
:End
:Opdracht

```

We bekijken twee voorbeelden.

De While-lus

Het programma vraagt een cijfer. Zolang het cijfer groter is dan -100 en kleiner is dan 100, wordt het getal verdubbeld. Dat cijfer wordt telkens getoond. Noem dit programma TEST4.

```
PROGRAM:TEST4
:Prompt D
:While D>-100 and
D<100
:2*D→D
:Disp "GETAL IS
NU",D
:End
```

```
:PROMPT D
:WHILE D>-100 and D<100
:2*D→D
:DISP "GETAL IS NU",D
:END
:DISP "GETAL IS FINAAL",D
```

Test je programma uit. Je zult merken dat het zo snel verloopt dat je de tussenliggende resultaten niet kunt zien. Dat kun je oplossen door voor END een PAUSE in te lassen. Dit heeft als gevolg dat telkens als D getoond wordt het programma even halt houdt totdat je weer op ENTER duwt.

Om een PAUSE in te lassen, ga je als volgt te werk:

1. Druk op **[PRGM]** > **[EDIT]**
2. Kies voor TEST4
3. Ga met je cursor vlak voor END staan en druk op **[2nd]** INS, gevolgd door **[ENTER]**. Je hebt nu een blanco lijn gemaakt.
4. Druk op **[PRGM]** > **[CTL]** en kies voor 8:PAUSE
5. Verlaat je programma via **[2nd]** QUIT.

Test het effect van je aanpassing.

Opmerking

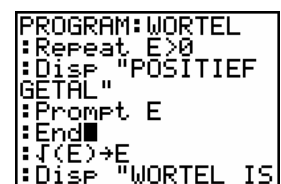
Programma's met programmalussen die fout geprogrammeerd zijn, kunnen wel eens oneindig lang doorgaan. Om een programma te onderbreken, druk je op **[ON]**. Vervolgens verschijnt het menu ERR:BREAK op het scherm. Kies 1:QUIT om terug te keren naar het basisscherm. Kies 2:GOTO om over te schakelen naar de programmaregel waar het programma werd onderbroken.

De Repeat-lus

Een Repeat-lus kan handig zijn als er voorwaarden gesteld dienen te worden aan de invoer van een programma.

We maken een programma dat alleen uitgevoerd wordt voor positieve getallen. Indien een negatief getal wordt ingevoerd, vraagt het programma een nieuw getal in te geven. Indien een positief getal wordt ingegeven berekent het programma de vierkantswortel. Noem het programma WORTEL.

```
:REPEAT E>0
:DISP "POSITIEF GETAL"
:PROMPT E
:End
:[√]E→E
:DISP "WORTEL IS",E
```



```
PROGRAM:WORTEL
:Repeat E>0
:Disp "POSITIEF
GETAL"
:Prompt E
:End
:√(E)→E
:Disp "WORTEL IS
```

Oefening 7

Pas het programma dat de BMI-waarde berekent zodanig aan dat de BMI-waarde pas berekend wordt als de ingegeven massa een positief getal is. Laat het programma ook antwoorden dat een BMI-waarde tussen de 20 en de 25 goed is. Een waarde die kleiner is dan 20, geeft aan dat de persoon te weinig weegt. Een waarde die groter is dan 25, geeft aan dat de persoon te veel weegt.

Werken met tellers: For-lus

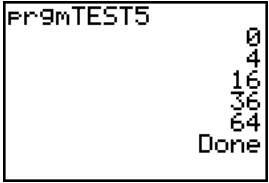
Je kunt een programmalus ook zo vaak laten uitvoeren als de gebruiker zelf wenst. In dat geval kun je gebruik maken van een For-lus. Bij een For-lus houdt een tellertje bij hoe vaak de lus al is uitgevoerd:

De juiste syntax is:

```
:For(teller,begin,einde,[stapgrootte])
:opdracht (zolang einde niet is overschreden)
:opdracht (zolang einde niet is overschreden)
:End
:opdracht
```

We maken een programma dat voor de even getallen tussen nul en acht de kwadraten toont op het scherm. Noem dit TEST5.

```
:FOR(A,0,8,2)
:DISP A²
:END
```



```
PrgrmTEST5
0
4
16
36
64
Done
```

Merk op dat je via de instructie DISP meteen het kwadraat van A kunt weergeven.

Oefening 8

Karl Gauss (van de “Gauss kromme” – 1777-1855) kreeg als zevenjarige de opdracht om de eerste honderd natuurlijke getallen bij elkaar op te tellen. Schrijf een programma dat voor jou deze klus klaart.

Oefening 9

Schrijf een programma dat de BAC-waarde berekent op basis van de nodige parameters. Het programma moet aangeven wanneer een parameter een niet toegelaten waarde bevat. Het programma moet ook advies geven over het al of niet besturen van de wagen. Wanneer de BAC waarde de 0,5 promille overschrijdt, mag een chauffeur niet rijden. Maak gebruik van een menustructuur.

$$BAC = \frac{n \cdot V \cdot p \cdot 8 \cdot 1,055}{r \cdot m} - t \cdot \beta$$

Hierin is

n: aantal gedronken glazen bier, wijn, alcoholpops...	$\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} n.V.p.8 \\ = \\ \text{aantal g zuivere alcohol} \end{array}$
V: volume per glas in cl	
p: alcoholpercentage	
1 cl alcohol = 8 g	

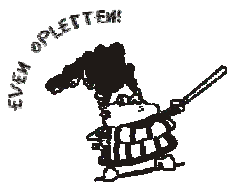
$$1 \text{ g bloed} = \frac{1}{1,055} \text{ ml bloed} = \frac{1}{1055} \text{ l bloed}$$

m: lichaamsmassa in kg

r-factor of Widmarkfactor: gemiddelde waarde is 0,68 voor mannen, 0,55 voor vrouwen

t: tijd in uren

β -factor of verteringsfactor met gemiddelde waarde $0,17 \frac{\text{g}}{\text{l.h}}$



Om een programma te verwijderen, druk je **[2nd]** MEM. Je kiest dan voor optie 2:Mem Mgmt/Del... en daarna voor optie 7:Prgm. Er verschijnt nu een lijst met alle programma's uit het geheugen. Door een programma aan te duiden en op **[DEL]** te drukken, kun je het programma verwijderen.

Om een programma naar een andere toestel te zenden, moet je beide toestellen verbinden met een link-kabeltje. Zet beide toestellen aan. Het ontvangende toestel zet je klaar om het programma te ontvangen via **[2nd]** LINK > RECIEVE **[ENTER]**. Druk bij het zendende toestel op **[2nd]** LINK > SEND > 3:Prgm. Maak een keuze uit het juiste programma, druk **[ENTER]** om te selecteren (één of meerdere programma's) en dan op TRANSMIT **[ENTER]**. Het programma wordt nu naar het andere toestel gezonden.

BIJLAGE

Bijkomende instructies

Input	In plaats van gebruik te maken van het commando PROMPT, kun je ook INPUT gebruiken om een invoer te vragen. Het voordeel van dit commando is dat je tekst en invoer op één regel kunt plaatsen (vb. Input "Massa= ",M). Lees hierover meer in de handleiding van je rekenmachine.
ClrTable	Met deze instructie kunt u tijdens de uitvoering van het programma het tabelscherm leeg maken.
Pause	Deze instructie onderbreekt de normale uitvoering van het programma, zodat je resultaten of grafieken kunt bekijken. Indien je op ENTER drukt, gaat de uitvoering verder.
Lbl, Goto	Lbl en Goto zijn instructies die u samen kunt gebruiken om sprongen binnen het programma uit te voeren.
DispGraph	Met deze instructie kunt u de huidige grafiek op het scherm tonen. U laat deze instructie best volgen door Pause.
DispTable	Met deze instructie kunt u de huidige tabel op het scherm tonen.
Graphstyle(	Definieert de grafiekstijl van een opgegeven functiegrafiek.