# Hoofdstuk 1

# Inleiding Maple

## 1.1 Inleiding

Mensen hebben door de eeuwen heen hulpmiddelen gebruikt voor het oplossen van wiskundige problemen. Schrijfgerei is natuurlijk altijd het belangrijkste gereedschap geweest. Maar daarnaast werden er apparaten ontwikkeld om het rekenen te ondersteunen. Het telraam (of abacus) was al in de Egyptische oudheid een erg populair rekenapparaat en werd tot voor kort in Rusland nog veelvuldig gebruikt. Vanaf de 17e eeuw werd de rekenliniaal ontwikkeld en werkte men aan de bouw van mechanische rekenmachines. En tegenwoordig is er de computer. Hedentendage wordt hij door veel mensen voornamelijk gebruikt ter vervanging van het schrijfgerei (tekstverwerking) maar je kunt er op wiskundig gebied nog veel meer interessante dingen mee doen. In de begintijd werd de computer vooral gebruikt om de mens grote numerieke problemen uit handen te nemen. Met de komst van *computeralgebra-systemen*, aan het einde van de 60-er jaren, blijkt de computer zelfs in staat tot differentiëren, integreren, vergelijkingen oplossen enz. Een computeralgebrasysteem is een groot programma dat in de computer geladen kan worden. Het biedt de mogelijkheid tot *interactief* gebruik: je geeft het systeem telkens een (wiskundige) opdracht, en het reageert er direct op. Een computeralgebra-systeem is in het algemeen in staat zowel met symbolen ($x$, $y$, $f(x)$) als met getallen te werken, en kan desgevraagd voor een wiskundig probleem op zoek gaan naar exacte oplossingen of naar numerieke benaderingen. Dit betekent zeker niet dat zo'n systeem wiskundigen overbodig maakt. Het is alleen een zeer ontwikkeld hulpmiddel waarmee routinematige berekeningen, zoals het bepalen van de afgeleide, aan de computer kunnen worden overgelaten, en dat gebruikt kan worden ter ondersteuning van wiskundig onderzoek.


Figuur 1.1: Het logo van Maple

Veel gebruikte systemen zijn op dit moment MACSYMA, Reduce, Derive, Mathematica, Axiom en Maple. Wij kozen voor Maple daar dit systeem zeer veel mogelijkheden heeft en door een groot aantal universiteiten in Nederland gebruikt wordt. De ontwikkeling van Maple begon eind 1980, aan de universiteit van Waterloo in Canada. Vandaar ook de naam: "maple" betekent esdoornblad, het symbool van de Canadese vlag. Sedert de eerste uitgave zijn er een groot aantal verbeteringen en uitbreidingen aan het systeem aangebracht en nieuwe uitgaven hebben elkaar in hoog tempo opgevolgd. De versie waarmee we tijdens het practicum aan de slag zullen gaan, is Maple 11, en wel in het bijzonder met de Classic Window versie. Deze versie is geïmplementeerd voor verschillende computersystemen, zoals Windows, Unix en Linux.

## 1.2 De New User's Tour

Maple start je via de All Programs-, Maple11- en Classic Worksheet Maple 11-button (**we gebruiken dus niet de gewone Maple 11 versie**). Er verschijnt een scherm waarin een cursor staat te knipperen. Dat heet een Maple worksheet. Binnen dit scherm voer je je commando's in en verschijnt de Maple output. Een worksheet kun je bewaren (via "File" en "Save As"), om later weer te kunnen gebruiken (via "File" en "Open"). Je kunt meer dan één worksheet tegelijk binnen Maple open hebben staan. Het verdient aanbeveling de worksheet tijdens het werk regelmatig te 'Save'n. Het gebeurt namelijk een enkele keer dat Maple crasht en al het werk verloren gaat, voorzover het niet ge'Save'd is.

Click nu op "Help" en vervolgens op "New User's Tour". Doorloop daarna onderdelen 1 (*Working Through the New User's Tour*) tot en met 6 (*Calculus*) en vervolgens het onderdeel 12 (*Help System*). Na ieder onderdeel kun je jezelf testen door onderstaande oefeningen te doen. Open daartoe een nieuw worksheet, en noem die hoofdstuk1. Zet daarin om te beginnen je naam, e-mail adres en studentnummer. Zet, waar nodig, commentaar bij je invoer of bij de output. De worksheet stuur je naar maple@few.vu.nl door de worksheet m.b.v. attach aan de e-mail te bevestigen. Dit verloopt in het begin niet altijd probleemloos. Stuur daarom de worksheet eerst naar jezelf om te zien of het goed gaat. Krijg je een e-mail vol met onbegrijpelijke code, dan is er iets mis gegaan. Het is de bedoeling dat degene die het werk nakijkt er doorheen kan gaan door alleen op ENTER te drukken, ga na of dat het geval is. Als het niet goed werkt, gaat het werk weer retour afzender. **Alleen opgaven die worden ingeleverd voor het tentamen van 20 oktober worden nagekeken. Bewaar je eigen werk zorgvuldig, e-mail is niet altijd betrouwbaar.**

### 1.2.1 Opgaven bij de "New User's Tour"

**Opgaven bij Topic** *Working Through the New User's Tour***:**

1. Beschrijf in je eigen woorden de betekenis van de begrippen 'Maple Input/Output', 'Execution Group', 'Worksheet' en 'Section'.

2. Oefen met het invoeren van tekst en formules. Zorg dat je commando's en formules kunt invoeren via Maple notatie (achter de cursor), in "standard math (input)" notatie (in tekst, zowel in zwart als in rood) en via "floating palettes" (via "View" en "Palettes").

3. Oefen met het maken van een (simpele) spreadsheet in Maple.
4. Oefen met 'Copy and Paste'.
5. Oefen in het gebruik van 'Context-sensitive Menus'.
6. Maak een nette indeling van een worksheet in sections, paragraphs, hyperlinks, etc. Gebruik tekst en Maple commando's.

**Opgaven bij Topic** *Numerical Calculations*:

1. Oefen met de commando's `ifactor` en `expand` (voor gehele getallen).
2. Geef een benadering van $\dfrac{2\sqrt{3}}{\sqrt{5}}$ in 20 decimalen.
3. Bereken $\displaystyle\sum_{n=0}^{N} \frac{2^n}{n!}$ voor $N = 10$ en $N = \infty$;
   Geef tevens een benadering van de uitkomsten in 20 decimalen.
4. Oefen met complexe getallen en het omzetten in modulus-argument notatie (pool-coördinaten).

**Opgaven bij Topic** *Algebraic Computations*:

1. Oefen met de commando's `expand`, `factor`, `normal` en `simplify` (nu voor algemene uitdrukkingen).
2. Definieer de functies $f$ en $g$ als $f(x) = x^3 - 2x^2 + x - 2$ en $g(x) = x^2 + 3x - 2$.
   Los vervolgens op: $f(x) = g(x)$. Benader daarna de 'wortel' oplossingen.
3. Los op: $\begin{cases} 2x + 3xy - y = 4, \\ x + 2y - 3xy = 0. \end{cases}$
4. Los op: $2x^2 + 4x \geq 3x + 6$.
5. Los op: $|x - 3| + x^2 = 4$.

**Opgaven bij Topic** *2-D Graphics*:

1. Oefen met de besproken commando's uit de pakketten `plots` en `plottools` (het is niet nodig je nu al in alle commando's te verdiepen). Let speciaal op het gebruik van het commando `display`: je kunt eerst een aantal commando's aan een variabele toekennen (dan een regel met '`:`' afsluiten!), om vervolgens met `display` de grafiek te tekenen (nu met '`;`' afsluiten).
2. Teken in één figuur, maar in verschillende kleuren, de grafieken van $\sin x$ en $\cos x$ op het interval $[-2\pi, 2\pi]$. Geef het plaatje een titel.
3. Teken de grafiek van $f(x) = x^3 - 3x$. Zet bij de toppen 'lokaal maximum' resp. 'lokaal minimum' (dit gaat via `textplot`).

**Opgaven bij Topic** *3-D Graphics*:

1. Teken in één figuur de grafieken van de functies $f(x,y) = 1 - x^2 - y^2$ en $g(x,y) = \sqrt{x^2+y^2}$.

2. Teken in één figuur het vlak met vergelijking $x+y+z = 0$ en de bol met middelpunt $(0,0,0)$ en straal 2 (dus $x^2+y^2+z^2 = 4$). Bekijk de figuur uit verschillende hoekpunten, door met de muis op de figuur te bewegen (houd daarbij de linker-muisknop vast).
   [Hint: gebruik het commando `implicitplot` in beide gevallen.]

**Opgaven bij Topic** *Calculus*:
N.B.: besteed niet te veel aandacht aan de diverse genoemde speciale functies. Het belangrijkste is dat je (piecewise gedefinieerde) functies kunt differentiëren en integreren en dat je een limiet kunt uitrekenen.

1. Laat Maple $\lim_{x\to\infty} \sin x$ bepalen. Wat vind je van het antwoord?

2. Bereken de limieten:
$$\lim_{x\to 0} \frac{\sin x}{\sqrt{x^2+x^3}}, \qquad \lim_{x\to 0^+} \frac{\sin x}{\sqrt{x^2+x^3}} \qquad \text{en} \qquad \lim_{x\to 0^-} \frac{\sin x}{\sqrt{x^2+x^3}}.$$

3. Bepaal van de functie $\sqrt{1+4x^2}$ de zevende afgeleide en vereenvoudig de output.

4. Bereken de integraal $\int x^3\sqrt{1+x^2}\,dx$. Controleer het antwoord door dat weer te differentiëren (vereenvoudig desnoods om te kunnen vergelijken!).

5. Voer in de functie
$$f(x) = \begin{cases} 0 & \text{als } x \le 0 \\ x^2 & \text{als } 0 < x < 1 \\ 2x-1 & \text{als } x \ge 1. \end{cases}$$
   Teken vervolgens de grafiek van $f$ op $[-2,2]$.

**Opgaven bij Topic** *Help System*:

1. Bekijk het Help menu en oefen met de diverse opties.

2. Zoek uit wat een 'partitie' is en bepaal vervolgens de partitie van 6.

3. Zoek uit met welk Maple-commando je kunt controleren of een getal een priemgetal is of niet. Ga vervolgens na of 1234567 een priemgetal is.

# Hoofdstuk 2

# Programmeren in Maple

## 2.1 Inleiding

Je hebt misschien al gemerkt dat Maple eigenlijk gewoon een hogere programmeertaal is. De algoritmes die we toepassen zitten echter verstopt en de gebruiker zal deze dan ook zelden of nooit te zien krijgen. Maple haalt ze eenvoudigweg even op als je een commando intikt, lost daarmee je probleem op (als je geluk hebt) en geeft het resultaat weer.

Het is mogelijk dat je niet geheel tevreden bent met sommige algoritmes. Je vindt het bijvoorbeeld slordig dat Maple de integratieconstante niet afdrukt bij het berekenen van primitieven. Je zou nu kunnen proberen op te zoeken hoe de procedure `int` precies in elkaar zit en daarin dan een wijziging proberen aan te brengen (doe dit nu niet, want het is nog een heel gedoe en de oorspronkelijke file is toch beschermd tegen indringers!). Ook kan het gebeuren dat je iets wilt laten uitrekenen, maar dat Maple niet over een procedure beschikt die dat voor je kan doen. Je zou nu zelf een programma kunnen schrijven, waarmee Maple je probleem kan oplossen. Als je dit programma hebt geschreven, dan laad je het in Maple, waarna Maple je hopelijk de juiste resultaten geeft.

Het is in deze cursus beslist *niet* de bedoeling om zelf nieuwe algoritmes te gaan schrijven, of om wijzigingen te gaan aanbrengen in bestaande Maple-algoritmes. Ten eerste is dit wiskundig gezien vaak een zeer moeilijke klus en ten tweede hebben we er geen tijd voor. Wat gaan we dan wel doen?

We zullen bestaande Maple algoritmes gaan inbedden in kleine programma's. We beperken ons daarbij tot enkele zeer eenvoudige procedures. Meer gedetailleerde informatie over programmeren in Maple kun je nalezen in Topic 10 van de New User's Tour.

## 2.2 De for-loop

De zogenaamde "for-loop"gebruik je als je dezelfde actie een aantal malen moet herhalen. Je geeft daarbij aan welke variabele je van waar tot waar laat lopen en welke opdracht er moet worden uitgevoerd. Bijvoorbeeld:

```
> f:=x->x^3-2*x+3;
> for i from 1 to 3 do diff(f(x),x$i) od;
```

$$f := x \to x^3 - 2\,x + 3$$

$$3\,x^2 - 2$$

$$6\,x$$

$$6$$

De variabele is hier dus de $i$ en die doorloopt de waarden 1, 2 en 3. Verder is hier het commando `do` belangrijk. Dit geeft aan dat er iets moet gebeuren. Als de for-loop is afgelopen, dan hoeft er ook niets meer te gebeuren, en wordt het `do` commando afgesloten met `od` (het omgekeerde van `do`!). De opdracht die Maple van ons krijgt is dus het berekenen van de eerste, tweede en derde afgeleide van de functie $f$ (let op hoe de $i$-de afgeleide wordt bepaald!). Merk op dat we de regels afsluiten met een punt-komma, omdat we alle berekende resultaten willen zien. In het algemeen is het handig om *niet* elke regel af te sluiten met een punt-komma, omdat we dan niet allerlei tussenresultaten (hier overigens niet aanwezig) afgedrukt krijgen.

We bekijken nog een voorbeeld. We tellen de eerste vijf *oneven* gehele getallen bijelkaar op en printen het eindresultaat:

```
> som:=0:
> for n from 1 to 9 by 2 do som:=som+n od:
> print(‘het totaal is‘):  som;
```

*het totaal is*

25

We gaan de stappen even na. Eerst definiëren we een variabele ‘som’ en stellen deze gelijk aan 0. Vervolgens stellen we de waarde van deze variabele steeds bij door er iedere keer n bij op te tellen. De variabele n loopt van 1 tot 9, maar maakt stapjes van 2 (`by 2`), zodat we achtereenvolgens 1, 3, 5, 7 en 9 bij ‘som’ hebben opgeteld. Daarna hebben we een printopdracht gegeven, waarvan de eerste slechts tekst afdrukt.

De output staat nu overigens niet op één regel. Hier is wel iets aan te doen, o.a. via het commando `lprint`, wat staat voor line-print (probeer de output nu eens op één regel te krijgen!).

**Opmerking 1:** Als je in een for-loop het commando `by k` niet toevoegt gaat Maple er vanuit dat je stapjes ter grootte 1 neemt. Overigens hoeft `k` niet geheel te zijn en zelfs niet positief! Test dit maar eens.

**Opmerking 2:** Het is verstandig af en toe een `restart` te geven. De lopende variabelen, zoals $i$ en $n$ in bovenstaande voorbeelden, hebben nu namelijk een waarde gekregen! Bedenk eerst eens welke waarden dit zijn en test dit vervolgens. Was je idee juist?

**Oefening:** Laat Maple door middel van een for-loop de volgende termen naast elkaar of onder elkaar afdrukken:

$$1, 1 + x, 1 + x + x^2, 1 + x + x^2 + x^3.$$

## 2.3 De while-loop

De "while-loop"gebruik je als je dezelfde actie een aantal maal wilt herhalen zolang er aan een bepaalde voorwaarde is voldaan. Een eenvoudig voorbeeld is het volgende

```
> x:=0:  while x^2<10 do print(2*x); x:=x+1:  od:
```

$$0$$
$$2$$
$$4$$
$$6$$

We begonnen met $x = 0$ en lieten daarna steeds $2x$ afdrukken, zolang $x^2$ maar kleiner was dan 10. Type dit zelf in en laat die regel uitvoeren. Vervang dan het dubbele-punt teken achter het commando `od:` door een punt-komma. Laat de regel nog eens uitvoeren. Verklaar het resultaat!

Een wat ingewikkelder voorbeeld is het volgende. Je leert bij het vak Calculus of bij het vak Differentiëren en Integreren dat (tot in het oneindige doorlopend)

$$1 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \ldots,$$

Je kunt dus 1 benaderen door voor een bepaalde $n \in \mathbb{N}$ uit te rekenen

$$\frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \ldots + \frac{1}{2^n}.$$

Je kunt je afvragen hoeveel termen je moet sommeren tot de uitkomst minder dan één miljardste ($= 10^{-9}$) van 1 afligt. Oftewel, hoe groot moeten we $n$ dan minimaal nemen? Je zou dan als volgt te werk kunnen gaan:

```
> n:=1:  som:=0:
> while evalf(1-som)>10^(-9) do
> som:=som+1/(2^n):  n:=n+1:  od:
> print('n'=n-1);print(evalf(som,15));
```

$$n = 30$$
$$0.99999999906867$$

Ga voor jezelf iedere stap nauwkeurig na. Als er iets niet duidelijk is, vraag het dan aan je practicumbegeleider. Merk op dat we niet $n$ maar $n - 1$ laten afdrukken [waarom doen we dat?].

**Oefening:** In bovenstaand voorbeeld wisten we al dat de uitkomst van de reeks 1 zou zijn en hebben we daar dankbaar gebruik van gemaakt. Dat is dus niet mogelijk als je niet weet wat het antwoord moet zijn. Deze oefening moet je daarom anders aanpakken!

We bekijken de rij $\{a_n\}$ gedefinieerd door

$$\begin{cases} a_{n+1} = \sqrt{1 + a_n}, & n \geq 1 \\ a_1 = 1, \end{cases}$$

Je kunt aantonen dat de limiet $\lim_{n\to\infty} a_n$ bestaat. Je moet hem nu met Maple bepalen via een iteratieproces. Gebruik als stopcriterium dat het verschil tussen twee opeenvolgende iteratiewaarden kleiner wordt dan $10^{-9}$.

## 2.4 Het if-statement

Een andere basis constructie is het "if-statement". Dit statement gebruik je als je eerst wilt controleren of aan een bepaalde conditie is voldaan, voordat je een bepaalde opdracht laat uitvoeren. In het algemeen ziet het if-statement er als volgt uit:

`if` (voorwaarde) `then`
    (een lijst Maple commando's)
`else`
    (een lijst Maple commando's)
`fi;`

Hierbij mag het `else` gedeelte worden weggelaten. Merk op dat het if-statement met `fi` eindigt. Iets dergelijks kwamen we al tegen bij `do` en `od`. We geven een voorbeeld:

```
> for i from 5 to 8 do
> if type(i,even) then print(i)
> else print(sqrt(i))
> fi
> od:
```

$$\sqrt{5}$$
$$6$$
$$\sqrt{7}$$
$$8$$

We hebben hierbij via het commando `type` laten nagaan of een getal even of oneven was. Aan de hand van de uitkomst moest er een waarde worden afgedrukt. Via de Help-functie kun je meer over het commando `type` te weten komen.

Je kunt verschillende condities tegelijkertijd testen via de logische operatoren `and` en `or`. In het volgende voorbeeld maken we gebruik van `and` en we vinden zo alle priemgetallen tot vijftig van de vorm $4n + 1$:

```
> for i from 1 to 50 do
> if isprime(i) and type((i-1)/4,integer)
> then print(i)
> fi:
> od:
```

$$5$$
$$13$$
$$17$$
$$29$$
$$37$$
$$41$$

**Oefening:** Beschouw het volgende wiskundige vraagstuk: $n$ is een geheel getal van drie cijfers, laten we zeggen $n = abc$, niet beginnend met een 0 (dus $a \neq 0$). Het getal $n$ heeft de eigenschap dat $n = a^3 + b^3 + c^3$. Een voorbeeld van een getal $n$ dat hieraan voldoet is 153. Immers: $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$. Bepaal nu *alle* mogelijke $n$ door een programmaatje te schrijven in Maple.

## 2.5 Procedures

In het inleidende hoofdstuk hebben we al gezien dat je in Maple functies kon definiëren met het symbool `->`. Aan zo'n functie konden we dan een naam toekennen, bijvoorbeeld $f$. Zo hadden we dan bijvoorbeeld via `f:=x->x^2` een functie gecreëerd, die aan ieder argument, als resultaat het kwadraat ervan retourneerde. In deze paragraaf zullen we een andere manier bespreken om functies te definiëren, waarbij we o.a. de `if, for` en `while` statements kunnen gebruiken.

Maple-gebruikers kunnen zelf nieuwe functies definiëren door gebruik te maken van de expressie `proc`, wat staat voor procedure. Alle Maple procedures zijn functies, waarin je een aantal waarden of namen als argument kunt meegeven. In de regel komt de procedure dan met een naam of waarde als resultaat. De procedure kun je uiteraard weer een naam geven. Laten we eens een eenvoudig voorbeeld bekijken (type dit over!).

We definiëren een functie van één variabele, die als output óf het kwadraat van de input geeft, indien deze negatief is, óf de wortel van de input in het andere geval:

```
> f:=proc(x)
> if x<0 then print(x^2)
> else print(sqrt(x)) fi
> end;
```

$$f := \mathbf{proc}(x)\,\mathbf{if}\,x < 0\,\mathbf{then}\,\mathrm{print}(x^2)\,\mathbf{else}\,\mathrm{print}(\mathrm{sqrt}(x))\,\mathbf{end\ if}\mathbf{end\ proc}$$

```
> f(5);f(-2.1);f(y);
```

$$\sqrt{5}$$

$$4.41$$

```
Error, (in f) cannot evaluate boolean: y < 0
```

Je ziet dat de functie goed werkt als we een getal als input geven, maar als we een letter invoeren, dan kan de procedure (terecht) niet beslissen of de input positief dan wel negatief is. Nu zien we snel in wat we fout gedaan hebben, maar als een programma erg groot wordt, dan is meestal niet makkelijk te achterhalen, waar de fout zit. Het is daarom handig om al op enkele foutmeldingen voorbereid te zijn, door vooraf even te checken of de input wel van de juiste soort is. Als je bijvoorbeeld in bovenstaand geval alleen maar getallen wilt als invoer (dus invoer moet van het type 'numeric' zijn, zie `?type`), dan kun je dat aan de procedure toevoegen volgens:

```
> f:=proc(x::numeric)
```

Doe dit nu ook in je worksheet en kijk wat Maple nu als output geeft. Een nog nettere manier is even in je procedure een extra check op de invoer te doen, waarna je via het commando `ERROR` precies kunt laten vertellen wat er fout is gegaan (pas je procedure weer aan!):

```
> f:=proc(x)
> if not type(x,numeric) then
> ERROR(‘De input moet een getal zijn.‘) fi:
> if x<0 then print(x^2)
> else print(sqrt(x)) fi
> end;
```

$f := \mathbf{proc}(x)$
**if not** type($x$, *numeric*) **then** ERROR(‘*De input moet een getal zijn.*‘) **end if**;
**if** $x < 0$ **then** print($x^2$) **else** print(sqrt($x$)) **end if**
**end proc**

```
> f(y);
```

```
Error, (in f) De input moet een getal zijn.
```

Tenslotte zullen we nog iets vertellen over locale en globale variabelen. Binnen je procedure is het vaak handig wat variabelen te hebben om mee te rekenen, maar die na afloop niet meer nodig zijn. Dit noemen we locale variabelen, en die kun je aan het begin van je procedure aangeven. Op deze manier kun je variabelen die je al eerder tijdens de Maple sessie een waarde had toegekend, opnieuw binnen je procedure gebruiken, zonder dat die eerdere waarde wordt overschreven of gebruikt. Een voorbeeld: we maken een procedure die de kwadraten van de eerste $n$ gehele getallen sommeert.

```
> k:=3;
```

$$k := 3$$

```
> kwadraatsom:=proc(n::integer)
> local k,som:
> som:=0; for k from 0 to n do som:=som+k^2 od:
> print(som):  end;
```

$kwadraatsom := \mathbf{proc}(n::integer)$
**local** $k$, $som$;
$som := 0$; **for** $k$ **from** 0 **to** $n$ **do** $som := som + k^2$ **end do**; print($som$)
**end proc**

```
> kwadraatsom(5);
```

$$55$$

```
> k;som;
```

$$3$$
$$som$$

De variabele $k$ heeft de waarde 3 op het moment dat we met de procedure `kwadraatsom` starten. Nu gaan we in de procedure de variabele $k$ weer gebruiken, maar we willen helemaal niet dat $k$ gelijk is aan 3, maar juist variabel lopend van 0 tot en met $n$. Dit doet Maple automatisch, maar er volgt wel een waarschuwing, als we dit niet expliciet hadden opgegeven (probeer dit maar eens door de regel `local k,som` weg te halen). Na afloop heeft $k$ nog steeds de waarde 3 en is er niets bekend over de variabele *som*.

Het is ook mogelijk de variabelen $k$ en *som* de waarden te laten houden die ze aan het eind van de procedure hebben gekregen. Dan moet je dit aangeven door ze als globale variabelen te declareren. Verander daartoe `local` in `global` en laat weer alle commando's uitvoeren. Je zult zien dat na afloop $k = 6$ en $som = 55$.

**Oefening (Het Syracuse probleem):** Het volgende beroemde onopgeloste probleem stamt uit de jaren 50 en staat bekend als het $3n + 1$-probleem of ook wel als het Syracuse probleem of als het Collatz probleem:

Ga uit van een positief geheel getal $n$. Als $n$ even is, dan deel je het getal door 2, anders vermenigvuldig je het met 3, tel je er 1 bij op, en deel je de uitkomst door 2. Kortom, je laat de functie $T$ los op $n$, waarbij $T$ gedefinieerd is als

$$T(n) = \begin{cases} \frac{1}{2}(3n+1) & \text{als } n \text{ oneven is} \\ \frac{1}{2}n & \text{als } n \text{ even is} \end{cases}$$

Vervolgens doe je met de uitkomst precies hetzelfde (je doet dan dus $T(T(n))$), enzovoorts. Een voorbeeld (ga na!). Start met $n = 20$, dan vind je achtereenvolgens:

$$20, 10, 5, 8, 4, 2, 1, 2, 1, 2, 1, \ldots.$$

Nu vermoedt men dat, als je maar lang genoeg doorgaat, je altijd op 1 uitkomt. Dit heeft men gecheckt tot alle getallen kleiner dan $10 \times 2^{58} = 2.882.303.761.517.117.440$, maar nog steeds heeft niemand bewezen dat het voor alle gehele positieve waarden van $n$ juist is.
De opdracht is nu, om een procedure te schrijven, die uitgaande van een door de gebruiker in te tikken waarde van $n$ het rijtje van $n$ tot en met 1 geeft (en dus niet doorgaat met $1, 2, 1, 2, \ldots$).
[Aanwijzing: definieer binnen een procedure eerst de functie $T$. Pas dan in een loop deze functie herhaaldelijk toe op $n, T(n), T(T(n)), \ldots$ tot de uitkomst 1 is.]

Ook nu moet je ter correctie weer een worksheet opsturen aan maple@few.vu.nl. Noem de worksheet 'hoofdstuk2' en begin weer met je naam, email-adres en studentnummer in te tikken. In de worksheet zet je de antwoorden op alle oefeningen van dit hoofdstuk.