

# Toets Programmeren, 2XP05 op 03-okt-2008, 09:00-12:00

TU/e Technische Universiteit Eindhoven  
Faculteit Wiskunde en Informatica

## PROBLEEM: Meer dan 100% inzet

Hoeveel is 100%? Wat betekent het om je meer dan 100% te geven? Heb je je ooit afgevraagd wat mensen bedoelen als ze zeggen dat ze zich meer dan 100% inzetten?

Ben je ook wel eens gevraagd meer dan 100% te presteren? Wat moet je doen om 105% te halen? Hoeveel is 100% eigenlijk?

Met wat wiskunde kun je daar eenvoudig achterkomen. Zoals gebruikelijk (in de numerologie en volgens de Kabbalah) heeft elke letter een waarde:  $a = 1$ ,  $b = 2$ ,  $c = 3$ , ...,  $z = 26$ . De **waarde van een frase** is de som van de waarden van de letters. Spaties en koppeltekens tellen *niet* mee.

Het is duidelijk dat je met alleen “**kennis**” =  $11 + 5 + 14 + 14 + 9 + 19 = 72$  niet aan de 100 komt. Ook “**ervaring**” =  $5 + 18 + 22 + 1 + 18 + 9 + 14 + 7 = 94$  is niet goed genoeg.

Met de juiste “**attitude**” =  $1 + 20 + 20 + 9 + 20 + 21 + 4 + 5 = 100$  lukt het net.

Terwijl “**niets doen**” =  $14 + 9 + 5 + 20 + 19 + 0 + 4 + 15 + 5 + 14 = 105$  ruim voldoet en met “**kletspraat**” =  $11 + 12 + 5 + 20 + 19 + 16 + 18 + 1 + 1 + 20 = 123$  kom je zelfs veel verder. Nog effectiever echter is “**manipuleren**” =  $13 + 1 + 14 + 9 + 16 + 21 + 12 + 5 + 18 + 5 + 14 = 128$ .

Schrijf een programma dat, gegeven een invoerbestand met frasen, een uitvoerbestand produceert met de bijbehorende waarden en wat statistieken.

## INVOER

Op standaardinvoer wordt een naamprefix (string van letters en/of cijfers) aangeboden, zeg *xyz*.

In het tekstbestand met naam *xyz.txt* (d.w.z. de naamprefix met daar achter *.txt*) staat op elke regel een frase bestaande uit kleine letters, d.w.z. karakters van 'a' t/m 'z' en mogelijk ook spaties.

## UITVOER

De uitvoer moet geschreven worden naar een nieuw aan te maken tekstbestand met naam *xyz-uit.txt*.

In het uitvoerbestand wordt voor iedere regel uit het invoerbestand een regel geschreven met daarop

- de waarde van de gelezen frase op 3 posities (links aangevuld spaties),
- gevolgd door één spatie,
- gevolgd door de frase.

Tenslotte wordt na al deze regels nog een drietal regels geschreven met daarop wat statistieken:

- eerst een regel met drie streepjes: ---

- dan het aantal frasen met waarde 100, gevolgd door één spatie en de tekst 'met waarde 100'
- tenslotte de tekst 'maximum' gevolgd door één spatie en de maximale waarde die vóórkomt.

Zie ook het voorbeeld.

**N.B.** Het maakt verder niet zo uit wat uw programma op standaarduitvoer schrijft. M.n. wordt een slotregel als 'Tik <return> ...' geaccepteerd.

## RESTRICTIES

U kunt er op rekenen dat het volgende geldt (uw programma hoeft dit dus niet te controleren):

- De opgegeven naamprefix is een geldig onderdeel van een bestandsnaam.
- Het opgegeven invoerbestand bestaat.
- Het opgegeven uitvoerbestand bestaat nog niet.
- Elke regel in het invoerbestand bevat één frase van minimaal één en maximaal 36 karakters, bestaande uit kleine letters ('a' t/m 'z'), spaties en/of koppeltekens. Er komen geen andere tekens in voor.
- Het invoerbestand bevat ten minste één frase.
- Het invoerbestand bevat minder dan 1000 (duizend) frasen.

## VOORBEELD

Table 1: :header-rows: 1

Standaard invoer	frasen.txt	frasen-uit.txt
frasen	kennis ervaring attitude niets doen kletspraat manipuleren	72 kennis 94 ervaring 100 attitude 105 niets doen 123 kletspraat 128 manipuleren --- 1 met waarde 100 maximum 128

## ANALYSE

Het programma kan het best opgebouwd worden uitgaande van de structuur van het invoerbestand.

Aangezien er geen speciale afsluitende regel wordt gebruikt, is het eerste patroon van het college niet van toepassing en dient *Eof* gebruikt te worden.

Aangezien de invoerregels kort zijn en een duidelijke structuur hebben, kunnen ze in hun geheel ingelezen en dan pas verwerkt worden, zoals in het tweede patroon het geval is. (Maar met het derde patroon kan het natuurlijk ook.) Het is niet aan te raden alle invoer op te slaan vóór verwerking.

Aangezien het analyseren van een frase met zich mee brengt dat de som over al de letters bepaald wordt, zal een geneste repetitie nodig zijn. **N.B.** Zelf-gedefinieerde routines zijn *niet* toegestaan.

## ONTWERP

Inventariseer welke stappen te onderscheiden zijn, welke variabelen daarvoor nodig zijn en wat hun rol is.

Bij de analyse hebben we al gezien dat er een geneste repetitie aan te pas komt. De “binnenste” repetitie heeft een duidelijke eigen taak: de waarde bepalen van de ingelezen frase. Daarbij speelt een beperkt stel variabelen een rol.

M.b.v. de standaardfunctie *ord* kan een teken (van een string ingelezen uit het invoerbestand) omgezet worden in een volgnummer. Via het verschil met *ord('a')* kun je de waarde van het betreffende teken bepalen (let op de +1). De letterwaarden zijn: a = 1, b = 2, c = 3, ..., z = 26.

## CODEREN

Het is verstandig niet het hele programma in één keer in te tikken en dan pas te controleren. Bouw het op in wat kleinere stappen.

1. Maak bijvoorbeeld eerst een programma dat de naam inleest, het invoerbestand opent en het uitvoerbestand creëert, de invoer regel voor regel doorleest en een kopie ervan in het uitvoerbestand schrijft. Controleer of dat goed werkt. Zorg meteen voor een nette opmaak, toepasselijk commentaar, en geschikte (constante) definities.
2. Voeg vervolgens het stukje programma toe dat de waarde van de ingelezen frase bepaalt en vóór de frase in het uitvoerbestand schrijft. Controleer ook dit.
3. Breid het programma nu uit met het tellen van aantal frasen met waarde 100 en schrijf de telling naar het uitvoerbestand. Controleer dit.
4. Voeg tenslotte de bepaling van het maximum toe.

Controleer of alle variabelen een geschikte *beginwaarde* krijgen. Denk ook aan de **codeerstandaard**.

**Normering:** Als van bovenstaande stappen alleen stap 1 gerealiseerd wordt, levert dat *geen* voldoende op. Met een programma dat t/m stap 2 realiseert, is maximaal een 8 te halen; t/m stap 3 maximaal een 9.

Het volstaat om alleen de Pascal programmatekst in te leveren. In Lazarus is dat het `.lpr` bestand.