

---

---

## EXAMEN: Computergebruik

---

---

Prof. Dr. Peter Dawyndt  
1<sup>e</sup> Bachelor Informatica  
groep 1

dinsdag 23-01-2006, 8:30h  
academiejaar 2006-2007  
eerste zittijd

### Opgave 1

In het bestand `geheim.txt` werd een geheime boodschap verborgen. Elke regel van dit bestand bevat een bitstring (een tekenreeks die enkel bestaat uit enen en nullen) van 16 karakters lang, gevolgd door een spatie en één enkel woord van willekeurige lengte, dat dus verder geen spaties meer bevat. Volg de onderstaande aanwijzingen op om deze boodschap te kunnen ontcijferen.

1. Bepaal reguliere expressies voor elk van de onderstaande verzamelingen, waarbij  $\mathcal{B}$  de verzameling van alle bitstrings voorstelt. Probeer deze reguliere expressies zo kort mogelijk te houden.
  - (a)  $A = \{x \in \mathcal{B} \mid \text{de eerste bit van } x \text{ is verschillend van de laatste bit}\}$
  - (b)  $B = \{x \in \mathcal{B} \mid \text{het aantal nullen in } x \text{ is een veelvoud van } 5\}$   
merk op: nul is deelbaar door 5
  - (c)  $C = \{x \in \mathcal{B} \mid \text{enen komen in } x \text{ steeds in paren voor}\}$   
voorbeeld: 0111101111111100 en 0110011001100110 behoren tot  $C$ , maar 0011101101101100 niet.

Gebruik het `egrep` commando om enkel die regels van het bestand `geheim.txt` te selecteren met een bitstring die tot een opgegeven verzameling behoort. Vermeld op je antwoordblad voor elke verzameling het gebruikte selectiecommando, en geef ook aan hoeveel regels je gevonden hebt.

2. De geheime boodschap bestaat uit drie woorden. Beschouw nu de verzamelingen  $A$ ,  $B$  en  $C$  zoals hierboven gedefinieerd. Het raadsel kan dan op de volgende manier worden opgelost:
  - (a) Het eerste woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap B$ .
  - (b) Het tweede woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap C$ .
  - (c) Het derde woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $B \cap C$ .

Vermeld op je antwoordblad voor elk gevonden woord het `egrep` commando (of de commandosequentie) die je gebruikt hebt om het woord te vinden. Geef ook de oplossing van het raadsel.

### Opgave 2

Gegeven is een bestand `marathons.txt`, waarvan de regels de volgende vorm hebben:

```
<datum> <atleet> <nationaliteit> <recordtijd> <land marathon>
```

Elke regel bevat informatie over een marathonrecord, waarbij een spatie wordt gebruikt als veldscheidingsteken. Dit betekent dus dat er geen spaties meer voorkomen in de inhoud van de velden zelf. Verder bevat het bestand ook nog commentaarregels die beginnen met een hekje (`#`).

Gevraagd wordt om, gebruik makend van de teksteditor `vi`, een reeks substitutiecommando's te geven die achtereenvolgens de volgende opdrachten uitvoeren.

1. Vervang datums van de vorm `d/mm/jjjj` door `0d/mm/jjjj`.
2. Verander de recordtijdnotatie van `u:mm:ss` naar `ss::mm::u`. Let op het aantal dubbelpunten dat hierbij gebruikt wordt.
3. Verwijder het veld `land marathon` (veld leegmaken, wel veldscheidingstekens behouden) uit alle records waar de atleet in zijn thuisland won.
4. Markeer alle records die gevestigd zijn in april of mei door een sterretje achteraan de lijn te plaatsen.
5. Reorganiseer het bestand zodat de records geordend zijn op nationaliteit en daarna op naam van de atleet. Zorg ervoor dat de regels na reorganisatie dezelfde vorm behouden. Het kan zijn dat je hierbij meer dan één commando nodig hebt.

Probeer zo weinig mogelijk commando's te geven en zo weinig mogelijk tekens te gebruiken in je commando's. De commentaarregels in `marathons.txt` mogen door je substituties niet gewijzigd worden. Alle wijzigingen moeten na elkaar uitgevoerd worden. Voeg de gebruikte substitutiecommando's toe aan je antwoordbestand.

### Opgave 3

Gegeven zijn de volgende tekstfragmenten:

1. **Lemma 1** *De formule*

$$\lim_{\alpha \rightarrow 0} \log_{\alpha} \left( 1 + \frac{(\alpha^x - 1)(\alpha^y - 1)}{\alpha - 1} \right) = \min\{x, y\}$$

is geldig voor alle  $(x, y) \in [0, 1]^2$ .

**Stelling 2** *Voor alle  $x$  en  $y$  in  $[0, 1]$  geldt:*

$$\lim_{\alpha \rightarrow 0} \log_{\alpha} \left( 1 + \frac{(\alpha^x - 1)(\alpha^y - 1)}{\alpha - 1} \right) = \begin{cases} x & \text{als } x \leq y \\ y & \text{anders} \end{cases}$$

2. Zij  $A$  en  $B$  verzamelingen in een universum  $X$ . Als  $\mu_A$  en  $\mu_B$  afbeeldingen zijn van  $X$  naar  $\{0, 1\}$  zodanig dat

$$(\forall x \in X)(\mu_A(x) = 1 \iff x \in A) \quad \text{en} \quad (\forall x \in X)(\mu_B(x) = 1 \iff x \in B),$$

dan geldt: 
$$\frac{|A \cap B|}{|A \cup B|} = \frac{\sum_{x \in X} \min\{\mu_A(x), \mu_B(x)\}}{\sum_{x \in X} (\mu_A(x) + \mu_B(x) - \min\{\mu_A(x), \mu_B(x)\})}.$$

3. De tabel

links		midden		rechts	
1	2	3		2	1
4	5	6		5	4
7	8	9		8	7

namaken in  $\LaTeX$  kan moeilijker zijn dan je denkt.

4. Stel dat  $T_{\mathbf{M}}$  en  $S_{\mathbf{M}}$  de  $[0, 1]^2 \rightarrow [0, 1]$  afbeeldingen zijn die gegeven worden door:

$$T_{\mathbf{M}}(x, y) = \min\{x, y\} \quad (1)$$

$$S_{\mathbf{M}}(x, y) = \max\{x, y\} \quad (2)$$

voor alle  $(x, y) \in [0, 1]^2$ . De gelijkheid  $S_{\mathbf{M}}(x, y) = 1 - T_{\mathbf{M}}(1 - x, 1 - y)$  geldt dan voor willekeurige  $x$  en  $y$  in  $[0, 1]$ , vermits:

$$\begin{aligned} 1 - T_{\mathbf{M}}(1 - x, 1 - y) &= 1 - \min\{1 - x, 1 - y\} && \text{want (1)} \\ &= 1 + \max\{-1 + x, -1 + y\} \\ &= \max\{1 - 1 + x, 1 - 1 + y\} \\ &= \max\{x, y\} \\ &= S_{\mathbf{M}}(x, y) && \text{want (2)} \end{aligned}$$

Gevraagd wordt om  $\text{\LaTeX}$ -code te genereren die precies hetzelfde resultaat oplevert als de tekstfragmenten binnen de bovenstaande kaders. Zorg ervoor dat zowel de  $\text{\LaTeX}$ -broncode die overeenstemt met elk tekstfragment als het eindresultaat zichtbaar zijn in de gecompileerde versie (PostScript of PDF) van je antwoordbestand.

## Opgave 4

Gegeven zijn de bestanden `TEX.txt`, `tex.txt`, `tex1.txt`, `tex2.txt`, `tex_c.txt`, `replace`, `ALPHABET.txt` en `sequences.txt`.

1. (a) Implementeer de volgende (eenvoudige) commando's aan de hand van shell files:
  - i. `./part <n>`  
Stuurt de standaard invoer naar standaard uitvoer vanaf de `<n>`-de regel. Met andere woorden, de eerste `<n> - 1` regels worden weggelaten.
  - ii. `./tolines`  
Zet alle karakters die binnenkomen via standaard invoer op een aparte regel, en stuurt het resultaat naar standaard uitvoer.
  - iii. `./tolower`  
Vervangt alle hoofdletters die binnenkomen via standaard invoer door kleine letters, en stuurt het resultaat naar standaard uitvoer.
- (b) Combineer deze commando's vervolgens tot (telkens) één samengesteld commando dat:
  - i. het bestand `TEX.txt` omzet naar `tex.txt`.
  - ii. het bestand `tex1.txt` genereert, vertrekkende van `TEX.txt`.
  - iii. het bestand `tex2.txt` genereert, vertrekkende van `tex1.txt`.
2. (a) Gebruik opnieuw shell files om de volgende (meer complexe) commando's te implementeren:
  - i. `./fuse <bestand 1> <bestand 2> ... <bestand n>`  
Plakt de overeenkomstige regels van de opgegeven bestanden en van standaard invoer aan elkaar, gescheiden door een TAB, en stuurt het resultaat naar standaard uitvoer.  
**Voorbeeld:** Stel dat de bestanden `1.txt` en `2.txt` beide slechts één regel bevatten, en dat die regels respectievelijk uit het karakter "1" en "2" bestaan (gevolgd door een newline). Het commando `./fuse 1.txt 2.txt` zal dan de regel "1 2" (en daarna een newline) uitschrijven. Als dit commando bovendien de regel "3" ontvangt via standaard invoer, dan wordt niet "1 2" maar "1 2 3" uitgeschreven.

- ii. `./sequences <bestand 1> <bestand 2> ... <bestand n>`

Creëert karaktersequenties door de overeenkomstige regels van de opgegeven bestanden aan elkaar te plakken, zonder scheidingstekens, en bepaalt vervolgens welke sequenties er het meeste opduiken. De 26 meest voorkomende karaktersequenties worden naar standaard uitvoer geschreven. Dit gebeurt onder de vorm van een lijst van sequenties die geordend is volgens aantal voorkomens. Met andere woorden, als een sequentie *a* vóór een andere sequentie *b* staat in de teruggegeven lijst, dan wil dat zeggen dat *a* meer voorkomt dan *b*.

**Voorbeeld:** Het gegeven bestand `sequences.txt` bevat de uitvoer van het commando `./sequences tex1.txt tex2.txt`.

- (b) De shell file `replace` krijg je cadeau. Deze file implementeert het commando `./replace <bestand>`, waarmee karaktersequenties in `<bestand>` kunnen vervangen worden. De regels die dit commando ontvangt via standaard invoer, moeten bestaan uit twee karaktersequenties gescheiden door een TAB. Elk voorkomen in `<bestand>` van de sequentie voor de TAB zal dan telkens vervangen worden door de sequentie erna. Het resultaat wordt uitgeschreven naar standaard uitvoer (en wordt dus niet opgeslagen in `<bestand>`).

Combineer `./replace` nu met `./fuse` en `./sequences` tot (telkens) één samengesteld commando dat:

- i. Het gegeven bestand `tex_c.txt` genereert, vertrekkende van `tex.txt`, `sequences.txt` en `ALPHABET.txt`. Dit houdt in dat de 26 meest voorkomende sequenties van twee karakters in `tex.txt` moeten vervangen worden door een hoofdletter. De sequentie die het meeste voorkomt wordt vervangen door "A", de op één na meest voorkomende door "B", etc.
- ii. Het bestand `tex.txt` genereert, vertrekkende van `tex_c.txt`, `tex1.txt`, `tex2.txt` en `ALPHABET.txt`.

Merk op dat `tex_c.txt` en `sequences.txt` samen 15% minder geheugenruimte innemen dan `tex.txt`, terwijl we `tex.txt` wel volledig kunnen reconstrueren uit `tex_c.txt` en `sequences.txt` (wetende dat we `ALPHABET.txt` moeten gebruiken bij die reconstructie). Met andere woorden, de hierboven besproken commando's implementeren een (zeer primitieve) vorm van compressie.

## Opgave 5

Een webfeed is een alternatieve, versimpelde weergave van online inhoud. Het is de meest gebruikte vorm van websyndicatie. Een site plaatst zijn inhoud, vaak in een verkorte of beschrijvende versie, in een online bestand: de *feed*. Veel sites kiezen ervoor om alleen nieuwskoppen, met een korte beschrijving erbij, in hun feed te zetten. Met behulp van een feed-lezer kun je de feed vervolgens lezen. Door een abonnement te nemen op meerdere webfeeds van verschillende aanbieders, kun je bijvoorbeeld in één oogopslag al het nieuws én alle nieuwe weblog-berichten overzien. De feed-lezer werkt dan de feeds op periodieke basis bij.

Really Simple Syndication (RSS) is een open XML-standaard voor dergelijke webfeeds. Een eenvoudig RSS2 bestand met 2 items ziet er als volgt uit:

```
<?xml version="1.0" ?>
<rss version="2.0">
<channel>

  <title>Titel website</title>
  <link>http://www.locatie_website.nl/</link>
  <description>beschrijving website</description>
```

```

<item>
  <title>titel item1</title>
  <link>http://www.locatie_website.nl/item1.html</link>
  <description>omschrijving item1</description>
</item>

<item>
  <title>titel item2</title>
  <link>http://www.locatie_website.nl/item2.html</link>
  <description>omschrijving item2</description>
</item>

</channel>
</rss>

```

Gevraagd wordt een eenvoudige feed-lezer te schrijven, onder de vorm van een shell script `fetchRSS` dat een URL als parameter meekrijgt die verwijst naar de locatie van een RSS-feed. De feed-lezer moet de RSS-webpagina lokaal binnenhalen en herformatteren naar XHTML-formaat, waardoor de feed kan bekeken worden in een webbrowser. Het volstaat hierbij om de `<title>` en `<link>` elementen die binnen een `<item>` element staan om te vormen tot een ongeordende lijst van titels die gelinkt zijn aan de gepaste URL. De weergave moet gebruik maken van twee verticale frames, waarbij de lijst van titels in het linkerframe wordt weergegeven. Als op een titel geklikt wordt, moet de webpagina van de gelinkte URL in het rechterframe getoond worden. Het commando

```
fetchRSS "http://rss.cnn.com/rss/cnn_tech.rss"
```

genereert dan XHTML bestanden, die als volgt in een webbrowser worden weergegeven:

The screenshot shows a CNN.com news page. On the left, there is a yellow sidebar with a list of links:

- [Microsoft to make Vista available online](#)
- [Yahoo to offer financial advice](#)
- [U.S.: Whoops! Report of spy coims not true](#)
- [Suit links MySpace to sex assaults](#)
- [Analyst: Apple at the top of its game](#)
- [Discount store's computer system hacked](#)
- [China: 2 million teens hooked on Web](#)
- [Commercial jet flies with anti-missile system](#)

The main content area shows the CNN.com logo and navigation menu. The featured article is titled "Microsoft to make Vista available online" and is dated "POSTED: 9:38 a.m. EST, January 19, 2007". The article includes a photo of a Windows Vista logo and text stating:

(AP) -- Microsoft Corp. will make its new Windows Vista operating system available for sale and download online, marking a new step for the software company, which has previously sold Windows only on packaged discs or pre-loaded on computers.

A relatively low number of computer users are likely to get Vista by downloading it from the Internet. But the mere availability indicates that Microsoft is fiddling with distribution methods for the extremely profitable franchise at the core of its business.

The download program, announced late Wednesday by the Redmond, Washington-based software maker, will

Bovendien moeten alle webpagina's die het script genereert voldoen aan de XHTML 1.0 standaard (controleer dit aan de hand van de W3C validator). Enkele andere voorbeelden van RSS-feeds zijn:

- BBC news ([http://newsrss.bbc.co.uk/rss/newsonline\\_world\\_edition/front\\_page/rss.xml](http://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_page/rss.xml))

- BBC sport ([http://newsrss.bbc.co.uk/rss/sportonline\\_world\\_edition/front\\_page/rss.xml](http://newsrss.bbc.co.uk/rss/sportonline_world_edition/front_page/rss.xml))
- De Standaard (<http://www.standaard.be/rss.aspx?section=Nieuws&subsection=Binnenland>)
- Humo ([http://www.humo.be/cps/rde/xchg/humo/rss\\_feed.xsl/rss\\_dezeweek.xml](http://www.humo.be/cps/rde/xchg/humo/rss_feed.xsl/rss_dezeweek.xml))
- Deus (<http://feeds.feedburner.com/TheEUSpodcast>)
- Science (<http://www.sciencemag.org/rss/ec.xml>)

**Hint:** Witruimte wordt genegeerd in XML-bestanden. Je gaat er dus best niet van uit dat de verschillende XML-elementen van de RSS-feed op een nieuwe regel beginnen.

**Hint:** Een XML-DTD schrijft geen vaste volgorde van kind-elementen voor.

---

---

## EXAMEN: Computergebruik

---

---

Prof. Dr. Peter Dawyndt  
1<sup>e</sup> Bachelor Informatica  
groep 2

dinsdag 23-01-2006, 13:30h  
academiejaar 2006-2007  
eerste zittijd

### Opgave 1

In het bestand `geheim.txt` werd een geheime boodschap verborgen. Elke regel van dit bestand bevat een bitstring (een tekenreeks die enkel bestaat uit enen en nullen) van 16 karakters lang, gevolgd door een spatie en één enkel woord van willekeurige lengte, dat dus verder geen spaties meer bevat. Volg de onderstaande aanwijzingen op om deze boodschap te kunnen ontcijferen.

1. Bepaal reguliere expressies voor elk van de onderstaande verzamelingen, waarbij  $\mathcal{B}$  de verzameling van alle bitstrings voorstelt. Probeer deze reguliere expressies zo kort mogelijk te houden.
  - (a)  $A = \{x \in \mathcal{B} \mid \text{het aantal nullen in } x \text{ is oneven}\}$
  - (b)  $B = \{x \in \mathcal{B} \mid \text{in } x \text{ komen maximaal twee opeenvolgende enen voor}\}$
  - (c)  $C = \{x \in \mathcal{B} \mid \text{de eerste en laatste bit van } x \text{ zijn } 0, \text{ en } x \text{ bevat geen opeenvolgende nullen}\}$

Gebruik het `egrep` commando om enkel die regels van het bestand `geheim.txt` te selecteren met een bitstring die tot een opgegeven verzameling behoort. Vermeld op je antwoordblad voor elke verzameling het gebruikte selectiecommando, en geef ook aan hoeveel regels je gevonden hebt.

2. De geheime boodschap bestaat uit drie woorden. Beschouw nu de verzamelingen  $A$ ,  $B$  en  $C$  zoals hierboven gedefinieerd. Het raadsel kan dan op de volgende manier worden opgelost:
  - (a) Het eerste woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap B$ .
  - (b) Het tweede woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap C$ .
  - (c) Het derde woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $B \cap C$ .

Vermeld op je antwoordblad voor elk gevonden woord het `egrep` commando (of de commandosequentie) die je gebruikt hebt om het woord te vinden. Geef ook de oplossing van het raadsel.

### Opgave 2

Gegeven is een bestand `dierengewicht.txt`, waarvan de regels de volgende vorm hebben:

```
<dier> <klasse> <min gew jong> <max gew jong> <min gew volw> <max gew volw>
```

Elke regel bevat informatie over het gewicht van een dier, waarbij een plusteken (+) wordt gebruikt als veldscheidingsteken. Dit betekent dus dat er geen plustekens meer voorkomen in de inhoud van de velden zelf. Gewichten hebben minimum 2 en maximum 3 cijfers na de komma. De gewichtsvelden hoeven niet noodzakelijk een waarde te hebben. Verder bevat het bestand ook nog commentaarregels die beginnen met een hekje (#).

Gevraagd wordt om, gebruik makend van de teksteditor `vi`, een reeks substitutiecommando's te geven die achtereenvolgens de volgende opdrachten uitvoeren.

1. Zorg dat alle gewichten drie cijfers na de komma hebben. Zo wordt bijvoorbeeld 0,20 aangevuld tot 0,200. Ontbrekende gewichten blijven onaangeroerd.
2. Markeer alle lijnen waarvoor het minimum gewicht van een jong niet gekend is. Doe dit door op het einde van de lijn de tekst //MISSING!// toe te voegen.
3. Verander de volgorde van de gewichten van

`<min gew jong> <max gew jong> <min gew volw> <max gew volw>`

naar

`<min gew jong> <min gew volw> <max gew jong> <max gew volw>.`

4. Reorganiseer het bestand op de volgende manier (na de aanpassing uit puntje 2 bevat het veld `<opmerking>` de tekst //MISSING!// indien het niet leeg is):

`<dier> <klasse> <min gew jong> <min gew volw> <max gew jong> <max gew volw> <opmerking>`

wordt omgezet naar

`<dier> MIN <min gewicht jong> <min gewicht volw>`

`<dier> MAX <max gewicht jong> <max gewicht volw>`

5. Orden de regels volgens jonge gewichten, je hoeft geen rekening te houden met minimum of maximum. De onderstaande regels staan bijvoorbeeld juist gesorteerd:

`dier_1 MIN 0.100 0.500`

`dier_1 MAX 0.200 0.800`

`dier_2 MIN 0.300 0.400`

`dier_2 MAX 0.350 0.500`

Zorg ervoor dat de regels na reorganisatie dezelfde vorm behouden. Het kan zijn dat je hiervoor meer dan één commando nodig hebt.

Probeer zo weinig mogelijk commando's te geven en zo weinig mogelijk tekens te gebruiken in je commando's. De commentaarregels in `dierengewicht.txt` mogen door je substituties niet gewijzigd worden. Alle wijzigingen moeten na elkaar uitgevoerd worden. Voeg de gebruikte substitutiecommando's toe aan je antwoordbestand.

## Opgave 3

Gegeven zijn de volgende tekstfragmenten:

1. **Lemma 1** *De formule*

$$\lim_{\gamma \rightarrow +\infty} \left[ 1 - \left( \frac{(1-x)^\gamma + (1-y)^\gamma}{2} \right)^{\frac{1}{\gamma}} \right] = \min\{x, y\}$$

*is geldig voor alle  $(x, y) \in [0, 1]^2$ .*

**Stelling 2** *Voor alle  $x$  en  $y$  in  $[0, 1]$  geldt:*

$$\lim_{\gamma \rightarrow +\infty} \left[ 1 - \left( \frac{(1-x)^\gamma + (1-y)^\gamma}{2} \right)^{\frac{1}{\gamma}} \right] = \begin{cases} x & \text{als } x \leq y \\ y & \text{anders} \end{cases}$$



2. Zij  $A$  en  $B$  verzamelingen in een universum  $X$ . Als  $\mu_A$  en  $\mu_B$  afbeeldingen zijn van  $X$  naar  $\{0, 1\}$  zodanig dat

$$(\forall x \in X)(\mu_A(x) = 1 \iff x \in A) \quad \text{en} \quad (\forall x \in X)(\mu_B(x) = 1 \iff x \in B),$$

dan geldt: 
$$\frac{2|A \cap B|}{|A| + |B|} = \frac{2 \sum_{x \in X} \max\{\mu_A(x) + \mu_B(x) - 1, 0\}}{\sum_{x \in X} (\mu_A(x) + \mu_B(x))}.$$

3. Stel dat  $T_{\mathbf{L}}$  en  $S_{\mathbf{L}}$  de  $[0, 1]^2 \rightarrow [0, 1]$  afbeeldingen zijn die gegeven worden door:

$$T_{\mathbf{L}}(x, y) = \max\{x + y - 1, 0\} \tag{1}$$

$$S_{\mathbf{L}}(x, y) = \min\{x + y, 1\} \tag{2}$$

voor alle  $(x, y) \in [0, 1]^2$ . De gelijkheid  $S_{\mathbf{L}}(x, y) = 1 - T_{\mathbf{L}}(1 - x, 1 - y)$  geldt dan voor willekeurige  $x$  en  $y$  in  $[0, 1]$ , vermits:

$$\begin{aligned} 1 - T_{\mathbf{L}}(1 - x, 1 - y) &= 1 - \max\{(1 - x) + (1 - y) - 1, 0\} && \text{want (1)} \\ &= 1 + \min\{-(1 - x) - (1 - y) + 1, 0\} \\ &= \min\{1 - 1 + x - 1 + y + 1, 1 + 0\} \\ &= \min\{x + y, 1\} \\ &= S_{\mathbf{L}}(x, y) && \text{want (2)} \end{aligned}$$

4. De tabel

<i>klein</i>		<i>groter</i>		<i>grootste</i>
1	2	3	4	5
10	20	30	40	50
100	200	300	400	500

namaken in  $\text{\LaTeX}$  kan moeilijker zijn dan je denkt.

Gevraagd wordt om  $\text{\LaTeX}$ -code te genereren die precies hetzelfde resultaat oplevert als de tekstfragmenten binnen de bovenstaande kaders. Zorg ervoor dat zowel de  $\text{\LaTeX}$ -broncode die overeenstemt met elk tekstfragment als het eindresultaat zichtbaar zijn in de gecompileerde versie (PostScript of PDF) van je antwoordbestand.

## Opgave 4

Gegeven zijn de bestanden `latex.txt`, `LATEX.txt`, `LATEX1.txt`, `LATEX2.txt`, `LATEX.c.txt`, `replace`, `alphabet.txt` en `chains.txt`.

1. (a) Implementeer de volgende (eenvoudige) commando's aan de hand van shell files:
  - i. `./caps`  
Vervangt alle kleine letters die binnenkomen via standaard invoer door hoofdletters, en stuurt het resultaat naar standaard uitvoer.
  - ii. `./drop <n>`  
Stuurt de standaard invoer naar standaard uitvoer vanaf de `<n>`-de regel. Met andere woorden, de eerste `<n> - 1` regels worden weggelaten.
  - iii. `./lines`  
Zet alle karakters die binnenkomen via standaard invoer op een aparte regel, en stuurt het resultaat naar standaard uitvoer.

- (b) Combineer deze commando's vervolgens tot (telkens) één samengesteld commando dat:
- i. het bestand `latex.txt` omzet naar `LATEX.txt`.
  - ii. het bestand `LATEX1.txt` genereert, vertrekkende van `latex.txt`.
  - iii. het bestand `LATEX2.txt` genereert, vertrekkende van `LATEX1.txt`.

2. (a) Gebruik opnieuw shell files om de volgende (meer complexe) commando's te implementeren:

- i. `./combine <bestand 1> <bestand 2> ... <bestand n>`

Plakt de overeenkomstige regels van standaard invoer en de opgegeven bestanden aan elkaar, gescheiden door een TAB, en stuurt het resultaat naar standaard uitvoer.

**Voorbeeld:** Stel dat de bestanden `b.txt` en `c.txt` beide slechts één regel bevatten, en dat die regels respectievelijk uit het karakter “b” en “c” bestaan (gevolgd door een newline). Het commando `./combine b.txt c.txt` zal dan de regel “b c” (en daarna een newline) uitschrijven. Als dit commando bovendien de regel “a” ontvangt via standaard invoer, dan wordt niet “b c” maar “a b c” uitgeschreven.

- ii. `./chains <bestand 1> <bestand 2> ... <bestand n>`

Creëert karaktersequenties door de overeenkomstige regels van de opgegeven bestanden aan elkaar te plakken, zonder scheidingstekens, en bepaalt vervolgens welke sequenties er het meeste opduiken. De 26 meest voorkomende karaktersequenties worden naar standaard uitvoer geschreven. Dit gebeurt onder de vorm van een lijst van sequenties die geordend is volgens aantal voorkomens. Met andere woorden, als een sequentie *a* vóór een andere sequentie *b* staat in de teruggegeven lijst, dan wil dat zeggen dat *a* meer voorkomt dan *b*.

**Voorbeeld:** Het bestand `chains.txt` bevat de uitvoer van het commando `./chains LATEX1.txt LATEX2.txt`.

- (b) De shell file `replace` krijg je cadeau. Deze file implementeert het commando `./replace <bestand>`, waarmee karaktersequenties in `<bestand>` kunnen vervangen worden. De regels die dit commando ontvangt via standaard invoer, moeten bestaan uit twee karaktersequenties gescheiden door een TAB. Elk voorkomen in `<bestand>` van de sequentie voor de TAB zal dan telkens vervangen worden door de sequentie erna. Het resultaat wordt uitgeschreven naar standaard uitvoer (en wordt dus niet opgeslagen in `<bestand>`).

Combineer `./replace` nu met `./combine` en `./chains` tot (telkens) één samengesteld commando dat:

- i. Het gegeven bestand `LATEX_c.txt` genereert, vertrekkende van `LATEX.txt`, `LATEX1.txt`, `LATEX2.txt` en `alphabet.txt`. Dit houdt in dat de 26 meest voorkomende sequenties van twee karakters in `LATEX.txt` moeten vervangen worden door een kleine letter. De sequentie die het meeste voorkomt wordt vervangen door “a”, de op één na meest voorkomende door “b”, etc.
- ii. Het gegeven bestand `LATEX.txt` genereert, vertrekkende van `LATEX_c.txt`, `chains.txt` en `alphabet.txt`.

Merk op dat `LATEX_c.txt` en `chains.txt` samen 12% minder geheugenruimte innemen dan `LATEX.txt`, terwijl we `LATEX.txt` wel volledig kunnen reconstrueren uit `LATEX_c.txt` en `chains.txt` (wetende dat we `alphabet.txt` moeten gebruiken bij die reconstructie). Met andere woorden, de hierboven besproken commando's implementeren een (zeer primitieve) vorm van compressie.

## Opgave 5

Een webfeed is een alternatieve, versimpelde weergave van online inhoud. Het is de meest gebruikte vorm van websyndicatie. Een site plaatst zijn inhoud, vaak in een verkorte of beschrijvende versie,

in een online bestand: de *feed*. Veel sites kiezen ervoor om alleen nieuwskoppen, met een korte beschrijving erbij, in hun feed te zetten. Met behulp van een feed-lezer kun je de feed vervolgens lezen. Door een abonnement te nemen op meerdere webfeeds van verschillende aanbieders, kun je bijvoorbeeld in één oogopslag al het nieuws én alle nieuwe weblog-berichten overzien. De feed-lezer werkt dan de feeds op periodieke basis bij.

Het *Atom Syndication Format* (of kortweg *Atom*) is een open XML-standaard voor dergelijke webfeeds. De ontwikkeling van Atom werd ingegeven door onvolkomenheden van het RSS 2.0 formaat. Atom is nu een officieel door de Internet Engineering Task Force voorgestelde standaard, zoals voorgeschreven in de recent uitgebrachte RFC 4287 (<http://www.ietf.org/rfc/rfc4287>). Een eenvoudig Atom-bestand met 3 entries ziet er als volgt uit:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<feed version="0.3" xmlns="http://www.w3.org/2005/Atom">

  <title>mozillaZine.org</title>
  <link rel="alternate" type="text/html" href="http://www.mozillazine.org/" />
  <modified>2007-01-20T01:31:48-08:00</modified>

  <entry>
    <title>SeaMonkey 1.1 Released</title>
    <link rel="alternate" type="text/html"
      href="http://www.mozillazine.org/talkback.html?article=20322" />
    <modified>2007-01-18T08:54:02-08:00</modified>
    <created>2007-01-18T08:54:02-08:00</created>
  </entry>

  <entry>
    <title>Mozilla Firefox 2.0.0.1 Released</title>
    <link rel="alternate" type="text/html"
      href="http://www.mozillazine.org/talkback.html?article=20238" />
    <modified>2006-12-24T15:33:11-08:00</modified>
    <created>2006-12-24T15:33:11-08:00</created>
  </entry>

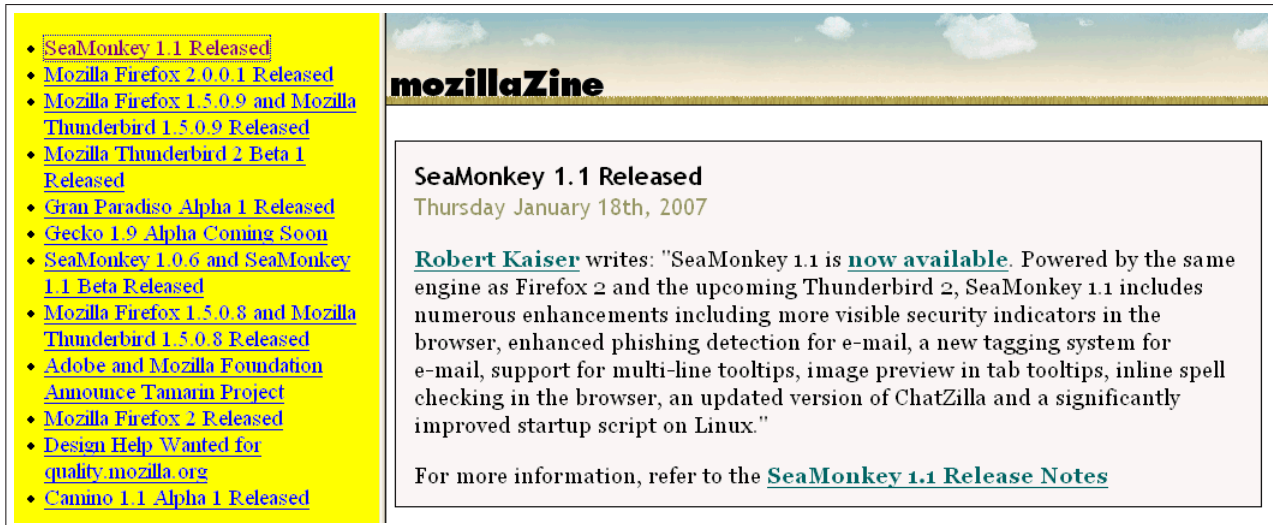
  <entry>
    <title>Mozilla Firefox 1.5.0.9 and Mozilla Thunderbird 1.5.0.9 Released</title>
    <link rel="alternate" type="text/html"
      href="http://www.mozillazine.org/talkback.html?article=20237" />
    <modified>2006-12-24T15:29:23-08:00</modified>
    <created>2006-12-24T15:29:23-08:00</created>
  </entry>

</feed>
```

Gevraagd wordt een eenvoudige feed-lezer te schrijven, onder de vorm van een shell script `fetchAtom` dat een URL als parameter meekrijgt die verwijst naar de locatie van een Atom-feed. De feed-lezer moet de Atom-webpagina lokaal binnenhalen en herformatteren naar XHTML-formaat, waardoor de feed kan bekeken worden in een webbrowser. Het volstaat hierbij om de `<title>` en `<link>` elementen die binnen een `<entry>` element staan om te vormen tot een ongeordende lijst van titels die gelinkt zijn aan de gepaste URL. De weergave moet gebruik maken van twee verticale frames, waarbij de lijst van titels in het linkerframe wordt weergegeven. Als op een titel geklikt wordt, moet de webpagina van de gelinkte URL in het rechterframe getoond worden. Het commando

fetchAtom "www.mozillazine.org/atom.xml"

genereert dan XHTML bestanden, die als volgt in een webbrowser worden weergegeven:



The screenshot shows the MozillaZine website. On the left, there is a yellow sidebar with a list of links. The main content area on the right features a header with the 'mozillaZine' logo and a featured article titled 'SeaMonkey 1.1 Released' dated Thursday January 18th, 2007. The article text, attributed to Robert Kaiser, describes the new features of SeaMonkey 1.1, such as enhanced security and phishing detection. A link to the 'SeaMonkey 1.1 Release Notes' is provided at the end of the article.

- [SeaMonkey 1.1 Released](#)
- [Mozilla Firefox 2.0.0.1 Released](#)
- [Mozilla Firefox 1.5.0.9 and Mozilla Thunderbird 1.5.0.9 Released](#)
- [Mozilla Thunderbird 2 Beta 1 Released](#)
- [Gran Paradiso Alpha 1 Released](#)
- [Gecko 1.9 Alpha Coming Soon](#)
- [SeaMonkey 1.0.6 and SeaMonkey 1.1 Beta Released](#)
- [Mozilla Firefox 1.5.0.8 and Mozilla Thunderbird 1.5.0.8 Released](#)
- [Adobe and Mozilla Foundation Announce Tamarin Project](#)
- [Mozilla Firefox 2 Released](#)
- [Design Help Wanted for \[quality.mozilla.org\]\(http://quality.mozilla.org\)](#)
- [Camino 1.1 Alpha 1 Released](#)

### SeaMonkey 1.1 Released

Thursday January 18th, 2007

**Robert Kaiser** writes: "SeaMonkey 1.1 is **now available**. Powered by the same engine as Firefox 2 and the upcoming Thunderbird 2, SeaMonkey 1.1 includes numerous enhancements including more visible security indicators in the browser, enhanced phishing detection for e-mail, a new tagging system for e-mail, support for multi-line tooltips, image preview in tab tooltips, inline spell checking in the browser, an updated version of ChatZilla and a significantly improved startup script on Linux."

For more information, refer to the [SeaMonkey 1.1 Release Notes](#)

Bovendien moeten alle webpagina's die het script genereert voldoen aan de XHTML 1.0 standaard (controleer dit aan de hand van de W3C validator). Kopieer voorbeelden van gegenereerde XHTML bestanden naar de diskette. Enkele andere voorbeelden van Atom-feeds zijn:

- Open Access News ([www.earlham.edu/~peters/fos/atom.xml](http://www.earlham.edu/~peters/fos/atom.xml))
- Blogzilla ([www.deftone.com/blogzilla/atom.xml](http://www.deftone.com/blogzilla/atom.xml))
- GoogleVideo ([googlevideo.blogspot.com/atom.xml](http://googlevideo.blogspot.com/atom.xml))
- Cites & Insights ([cical.blogspot.com/atom.xml](http://cical.blogspot.com/atom.xml))
- The Django Book ([www.djangobook.com/feed/](http://www.djangobook.com/feed/))

**Hint:** Witruimte wordt genegeerd in XML-bestanden. Je gaat er dus best niet van uit dat de verschillende XML-elementen van de Atom-feed op een nieuwe regel beginnen.

**Hint:** Een XML-DTD schrijft geen vaste volgorde van kind-elementen voor.

---

---

## EXAMEN: Computergebruik

---

---

Prof. Dr. Peter Dawyndt  
1<sup>e</sup> Bachelor Informatica  
groep 3

woensdag 24-01-2006, 8:30h  
academiejaar 2006-2007  
eerste zittijd

### Opgave 1

In het bestand `geheim.txt` werd een geheime boodschap verborgen. Elke regel van dit bestand bevat een bitstring (een tekenreeks die enkel bestaat uit enen en nullen) van 16 karakters lang, gevolgd door een spatie en één enkel woord van willekeurige lengte, dat dus verder geen spaties meer bevat. Volg de onderstaande aanwijzingen op om deze boodschap te kunnen ontcijferen.

1. Bepaal reguliere expressies voor elk van de onderstaande verzamelingen, waarbij  $\mathcal{B}$  de verzameling van alle bitstrings voorstelt. Probeer deze reguliere expressies zo kort mogelijk te houden.
  - (a)  $A = \{x \in \mathcal{B} \mid \text{het aantal nullen in } x \text{ is even}\}$
  - (b)  $B = \{x \in \mathcal{B} \mid \text{de reeks } 01 \text{ komt in } x \text{ evenveel voor als de reeks } 10\}$
  - (c)  $C = \{x \in \mathcal{B} \mid x \text{ bevat geen opeenvolgende enen}\}$

Gebruik het `egrep` commando om enkel die regels van het bestand `geheim.txt` te selecteren met een bitstring die tot een opgegeven verzameling behoort. Vermeld op je antwoordblad voor elke verzameling het gebruikte selectiecommando, en geef ook aan hoeveel regels je gevonden hebt.

2. De geheime boodschap bestaat uit drie woorden. Beschouw nu de verzamelingen  $A$ ,  $B$  en  $C$  zoals hierboven gedefinieerd. Het raadsel kan dan op de volgende manier worden opgelost:
  - (a) Het eerste woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap B$ .
  - (b) Het tweede woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap C$ .
  - (c) Het derde woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $B \cap C$ .

Vermeld op je antwoordblad voor elk gevonden woord het `egrep` commando (of de commandosequentie) die je gebruikt hebt om het woord te vinden. Geef ook de oplossing van het raadsel.

### Opgave 2

Gegeven is een bestand `overschrijvingen.txt`, waarvan de regels de volgende vorm hebben:

```
<rekening opdrachtgever> <rekening begunstigde> <opmerking> <bedrag> <OGM> <VM> <memodatum>
```

Elke regel bevat informatie over een banktransactie, waarbij een plusteken (+) wordt gebruikt als veldscheidingsteken. Dit betekent dus dat er geen plustekens meer voorkomen in de inhoud van de velden zelf. Rekeningnummers zijn van de vorm `xxx-xxxxxxx-xx`, waarbij `x` voor een willekeurig cijfer staat. De memodatum is van de vorm `dd/mm/jjjj`. Het `OGM`-veld is een gestructureerde mededeling, en heeft de volgende vorm: `***xxx/xxxx/xxxxxxx***`, waarbij `x` opnieuw voor een willekeurig cijfer staat. Je mag ervan uitgaan dat enkel het `OGM`-veld sterretjes kan bevatten. De gestructureerde mededeling en de vrije mededeling (`VM`, willekeurige opeenvolging van tekens) moeten niet noodzakelijk ingevuld

worden. Het opmerkingsveld geeft aan of het om een overschrijving tussen eigen rekeningen (**EIGEN**) gaat of niet (**ANDER**). Verder bevat het bestand ook nog commentaarregels die beginnen met een hekje (#).

Gevraagd wordt om, gebruik makend van de tekeditor vi, een reeks substitutiecommando's te geven die achtereenvolgens de volgende opdrachten uitvoeren.

1. Vul het VM-veld met de tekst **Eigen rekeningen**, en dit enkel voor de transacties tussen eigen rekeningen waarvoor het OGM-veld en het VM-veld beide geen waarde bevatten.
2. Verwijder het opmerkingsveld.
3. Verander de notatie van **\*\*\*abc/defg/hijkl\*\*\*** naar **\*\*\*defg/hijkl/abc\*\*\*** voor het OGM-veld.
4. Markeer alle transacties met een bedrag dat 1000 euro of meer bedraagt. Doe dit door op het einde van de lijn de tekst **HOOG!** te zetten.
5. Reorganiseer het bestand, zodanig dat alle transacties geordend zijn volgens het rekeningnummer van de begunstigde. Zorg ervoor dat het "hoogste" rekeningnummer (m.a.w. een rekeningnummer dat begint met een 9 is hoger dan een rekeningnummer dat begint met een 8) bovenaan staat. Het kan zijn dat je hiervoor meer dan één commando nodig hebt.

Probeer zo weinig mogelijk commando's te geven en zo weinig mogelijk tekens te gebruiken in je commando's. De commentaarregels in **overschrijvingen.txt** mogen door je substituties niet gewijzigd worden. Alle wijzigingen moeten na elkaar uitgevoerd worden. Voeg de gebruikte substitutiecommando's toe aan je antwoordbestand.

### Opgave 3

Gegeven zijn de volgende tekstfragmenten:

1. **Lemma 1** *De formule*

$$\lim_{\lambda \rightarrow +\infty} \sqrt[\lambda]{\max\{1 - [(1-a)^\lambda + (1-b)^\lambda], 0\}} = \min\{a, b\}$$

is geldig voor alle  $(a, b) \in [0, 1]^2$ .

**Stelling 2** *Voor alle  $a$  en  $b$  in  $[0, 1]$  geldt:*

$$\lim_{\lambda \rightarrow +\infty} \sqrt[\lambda]{\max\{1 - [(1-a)^\lambda + (1-b)^\lambda], 0\}} = \begin{cases} a & \text{indien } a < b \\ b & \text{anders} \end{cases}$$

2. Zij  $U$  een universum,  $X \subseteq U$  en  $Y \subseteq U$ . Als  $\mu_X$  en  $\mu_Y$  afbeeldingen zijn van  $U$  naar  $\{0, 1\}$  zodanig dat

$$(\forall u \in U)(\mu_X(u) = 0 \iff u \notin X) \quad \text{en} \quad (\forall u \in U)(\mu_Y(u) = 0 \iff u \notin Y),$$

dan geldt: 
$$\frac{2|X \cap Y|}{|X \setminus Y| + |Y \setminus X| + 2|X \cap Y|} = \frac{2 \sum_{u \in U} \mu_X(u) \cdot \mu_Y(u)}{\sum_{u \in U} (\mu_X(u) + \mu_Y(u))}.$$

3. Beschouw de  $[0, 1]^2 \rightarrow [0, 1]$  afbeeldingen  $T_{\mathbf{P}}$  en  $S_{\mathbf{P}}$  die gegeven worden door:

$$T_{\mathbf{P}}(a, b) = a \cdot b \quad (1)$$

$$S_{\mathbf{P}}(a, b) = 1 - T_{\mathbf{P}}(1 - a, 1 - b) \quad (2)$$

voor alle  $(a, b) \in [0, 1]^2$ . De gelijkheid  $S_{\mathbf{P}}(a, b) = a + b - a \cdot b$  geldt voor willekeurige  $a$  en  $b$  in  $[0, 1]$ , vermits:

$$\begin{aligned} S_{\mathbf{P}}(a, b) &= 1 - T_{\mathbf{P}}(1 - a, 1 - b) && \text{want (2)} \\ &= 1 - ((1 - a) \cdot (1 - b)) && \text{want (1)} \\ &= 1 - (1 - a - b + a \cdot b) \\ &= 1 - 1 + a + b - a \cdot b \\ &= a + b - a \cdot b \end{aligned}$$

4. De volgende tabel geeft een overzicht van de meest recente versienummers van het softwarepakket  $\text{T}_{\text{E}}\text{X}$ :

versienummers $\text{T}_{\text{E}}\text{X}$ :	gebruikelijke nummers:
3.1415	3.4
3.14159	3.5
3.141592	3.6

Zoals blijkt uit deze tabel, naderen de versienummers van  $\text{T}_{\text{E}}\text{X}$  naar  $\pi$ .

Gevraagd wordt om  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -code te genereren die precies hetzelfde resultaat oplevert als de tekstfragmenten binnen de bovenstaande kaders. Zorg ervoor dat zowel de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -broncode die overeenstemt met elk tekstfragment als het eindresultaat zichtbaar zijn in de gecompileerde versie (PostScript of PDF) van je antwoordbestand.

## Opgave 4

Gegeven zijn de volgende bestanden: `PANGRAM.txt`, `pangram.txt`, `letters1.txt`, `letters2.txt`, `PANGRAM_e.txt`, `substitute`, `alphabet.txt` en `key.txt`.

1. (a) Implementeer de volgende (eenvoudige) commando's aan de hand van shell files:
  - i. `./letters`  
Verwijdert alle karakters die geen letter zijn uit standaard invoer en schrijft het resultaat naar standaard uitvoer. Met andere woorden, alles wat geen letter is wordt weggefilterd.
  - ii. `./lowercase`  
Vervangt alle hoofdletters die binnenkomen via standaard invoer door kleine letters, en stuurt het resultaat naar standaard uitvoer.
  - iii. `./separate`  
Zet alle karakters die binnenkomen via standaard invoer op een aparte regel, en stuurt het resultaat naar standaard uitvoer.
- (b) Combineer deze commando's vervolgens tot (telkens) één samengesteld commando dat:
  - i. het bestand `PANGRAM.txt` omzet naar `pangram.txt`.
  - ii. het bestand `letters1.txt` genereert, vertrekkende van `pangram.txt`.
  - iii. het bestand `letters2.txt` genereert, vertrekkende van `PANGRAM.txt`.
2. (a) Gebruik opnieuw shell files om de volgende (meer complexe) commando's te implementeren:

- i. `./couple <bestand 1> <bestand 2> ... <bestand n>`

Plakt de overeenkomstige regels van de opgegeven bestanden en van standaard invoer aan elkaar, gescheiden door een TAB, en stuurt het resultaat naar standaard uitvoer.

**Voorbeeld:** Stel dat de bestanden `a.txt` en `b.txt` beide slechts één regel bevatten, en dat die regels respectievelijk uit het karakter “a” en “b” bestaan (gevolgd door een newline). Het commando `./couple a.txt b.txt` zal dan de regel “a b” (en daarna een newline) uitschrijven. Als dit commando bovendien de regel “c” ontvangt via standaard invoer, dan wordt niet “a b” maar “a b c” uitgeschreven.

- ii. `./genkey <bestand 1> <bestand 2> ... <bestand n>`

Stuurt een lijst van karaktersequenties naar standaard uitvoer, waarbij elke karaktersequentie overeenkomt met de inhoud van minstens één regel in één van de opgegeven bestanden. Als een karaktersequentie *a* vóór een andere karaktersequentie *b* staat in de teruggegeven lijst, dan wil dat zeggen dat *a* minder voorkomt dan *b*. Met andere woorden, het eerste karakter komt het minste voor, en het laatste het meeste.

**Voorbeeld:** Het bestand `key.txt` bevat de uitvoer van het commando `./genkey letters2.txt`.

- (b) De shell file `substitute` krijg je van ons cadeau. Deze file implementeert het commando `./substitute <bestand>`, waarmee karakters in `<bestand>` kunnen vervangen worden. De regels die dit commando ontvangt via standaard invoer, moeten bestaan uit twee karakters gescheiden door een TAB. Elk voorkomen in `<bestand>` van het karakter voor de TAB zal dan telkens vervangen worden door het karakter erna. Het resultaat wordt uitgeschreven naar standaard uitvoer (en wordt dus niet opgeslagen in `<bestand>`).

Combineer `./substitute` nu met `./couple` en `./genkey` tot (telkens) één samengesteld commando dat:

- i. Het bestand `PANGRAM.e.txt` genereert, vertrekkende van de bestanden `PANGRAM.txt`, `letters 2.txt` en `alphabet.txt`. Dit houdt in dat elke letter in `PANGRAM.txt` moet vervangen worden door een andere letter. De letter “a” wordt vervangen door de letter die het minste voorkomt, “b” door de op één na minst voorkomende letter, etc.
- ii. Het bestand `PANGRAM.txt` genereert, vertrekkende van `PANGRAM.e.txt`, `key.txt` en `alphabet.txt`.

Merk op dat `PANGRAM.e.txt` onleesbaar is, en dat het behoorlijk moeilijk is om het om te zetten naar `PANGRAM.txt` als je niet over `key.txt` beschikt (en ook niet weet hoe je `key.txt` moet genereren). Met andere woorden, de hierboven besproken commando’s implementeren een (zeer primitieve) vorm van encryptie.

## Opgave 5

Veel gegevens bevatten een ruimtelijke component, waardoor we die gegevens een plaats op een kaart kunnen geven. Het proces waarbij een object op een kaart wordt geplaatst, wordt *geocoderen* genoemd. Een *geocoder* is dan weer een stuk software of een webservice die helpt in dit proces. In deze opgave zullen we gebruik maken van zo een geocoder webservice voor het opvragen van geografische informatie die we op een website zullen plaatsen.

1. De broncode van de website <http://brainoff.com/geocoder/countryselect.php> bevat een lijst van landen, en hun ISO 3166-1 landcode die bestaat uit twee letters (de landcode van België is bijvoorbeeld `be`). Gevraagd wordt om op basis van deze lijst een shell script `countryCode` te schrijven, dat de Engelstalige naam van een land als argument meekrijgt, en de corresponderende landcode uitschrijft naar standaard uitvoer. Zorg ervoor dat het shell script niet hoofdlettergevoelig is wat betreft het doorgegeven argument. Hierbij is het toegelaten dat het shell script automatisch de broncode van de opgegeven website lokaal kopieert, op voorwaarde dat deze



tijdelijke kopie bij het afsluiten van het shell script terug verwijderd wordt. Enkele voorbeelden ter controle:

```
./countryCode China      -->   ch
./countryCode PERU       -->   pe
./countryCode latvia     -->   lg
```

Let er dus op dat de landcode niet altijd gelijk is aan de eerste twee letters van de landnaam.

2. Gebruik het shell script `countryCode` bij het laten genereren van een webpagina door een shell script `showInfo`. Dit shell script krijgt als argumenten de Engelstalige naam van een stad en een land mee. Zo genereert het commando

```
./showInfo Ghent Belgium
```

bijvoorbeeld een website die gebruik maakt van frames, en die er in een browser ongeveer uitziet zoals in de onderstaande figuur staat weergegeven.



The screenshot shows the Wikipedia article for Ghent, Belgium. The title is "Ghent, Belgium (51.05,3.71) (NL, FR, EN)". The article text describes Ghent as the capital of the Flemish province of East Flanders and the capital of the Arrondissement of Ghent. It mentions the city's location on the Scheldt and Leie rivers, its population of approximately 233,120 in 2006, and its historical significance as the *Arteveldestad*. The article also mentions the city's patron saints, the HH. Lieven en Pharaïldis, and the city's coat of arms and flag. A map of the province of East Flanders is shown, with Ghent highlighted in red. The article is part of a frame-based website, with a navigation menu on the left and a search bar at the bottom.

Het bovenste frame bevat de naam van de stad en het land (gescheiden door een komma) die als argumenten aan het shell script werden doorgegeven, gevolgd door de coördinaten van de stad (breedte- en lengtegraden). Deze coördinaten kunnen worden opgevraagd via de webservice

```
http://brainoff.com/geocoder/rest/?city=
```

Achter deze template-URL moet de naam van een stad en de landcode worden toegevoegd, gescheiden door een komma. Voor ons voorbeeld van de stad Ghent wordt dit

```
http://brainoff.com/geocoder/rest/?city=Ghent,be
```

Deze webservice geeft dan een XML-bestand terug, waaruit de coördinaten kunnen gefilterd worden. Voor ons voorbeeld ziet dit XML-bestand er als volgt uit:

```

<rdf:RDF xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <geo:Point>
    <geo:long>3.7166667</geo:long>
    <geo:lat>51.05</geo:lat>
  </geo:Point>
</rdf:RDF>

```

Let erop dat de lengte- en breedtegraden in het XML-bestand precies in de omgekeerde volgorde staan dan deze waarin ze moeten worden weergegeven in de gegenereerde website. Zorg er ook voor dat op de webpagina enkel twee cijfers na de komma van de coördinaten worden weergegeven.

Na de coördinaten bevat het bovenste frame ook nog drie links naar de Nederlandstalige (NL), Franstalige (FR) en Engelstalige (EN) wikipagina's van de stad die als parameter werd doorgegeven aan het shell script. Voor de stad Gent zien de URLs van deze wikipagina's er respectievelijk als volgt uit:

```

href=http://nl.wikipedia.org/wiki/Ghent
href=http://fr.wikipedia.org/wiki/Ghent
href=http://en.wikipedia.org/wiki/Ghent

```

waarbij het laatste stuk van de URLs dus telkens bestaat uit de naam van de stad die als parameter aan het shell script werd doorgegeven. Bij het aanklikken van de hyperlinks (NL, FR, EN) moet de webpagina in het onderste frame getoond worden. Zorg er ook voor dat de Nederlandstalige wikipagina van de opgegeven stad standaard getoond wordt in het onderste frame. Bovendien moeten alle webpagina's die het script genereert voldoen aan de XHTML 1.0 standaard (controleer dit aan de hand van de W3C validator). Kopieer voorbeelden van gegenereerde XHTML bestanden naar de USB-stick. Je kan het resultaat testen aan de hand van de volgende voorbeelden

```

./showInfo Ghent Belgium
./showInfo Brussels Belgium
./showInfo Lima Peru
./showInfo Beijing China

```

**Hint:** Je kan backquotes gebruiken om commando-substitutie uit te voeren. Bij het commando

```
echo landcode: './countryCode Belgium'
```

zal eerst het stuk tussen de backquotes worden uitgevoerd (`./countryCode Belgium`), en wordt daarna de uitvoer op de commandolijn gesubstitueerd. De commando-lijn wordt dus `echo landcode: be`, waardoor dus de tekst `landcode: be` naar standaard uitvoer wordt geschreven.

---

---

## EXAMEN: Computergebruik

---

---

Prof. Dr. Peter Dawyndt  
1<sup>e</sup> Bachelor Informatica  
groep 4

woensdag 24-01-2006, 13:30h  
academiejaar 2006-2007  
eerste zittijd

### Opgave 1

In het bestand `geheim.txt` werd een geheime boodschap verborgen. Elke regel van dit bestand bevat een bitstring (een tekenreeks die enkel bestaat uit enen en nullen) van 16 karakters lang, gevolgd door een spatie en één enkel woord van willekeurige lengte, dat dus verder geen spaties meer bevat. Volg de onderstaande aanwijzingen op om deze boodschap te kunnen ontcijferen.

1. Bepaal reguliere expressies voor elk van de onderstaande verzamelingen, waarbij  $\mathcal{B}$  de verzameling van alle bitstrings voorstelt. Probeer deze reguliere expressies zo kort mogelijk te houden.
  - (a)  $A = \{x \in \mathcal{B} \mid \text{de laatste drie bits van } x \text{ zijn gelijk}\}$
  - (b)  $B = \{x \in \mathcal{B} \mid \text{het aantal nullen in } x \text{ is deelbaar door } 3\}$   
merk op: nul is deelbaar door 3
  - (c)  $C = \{x \in \mathcal{B} \mid \text{als } 000 \text{ een substring is van } x, \text{ dan is } n_0(x) = 3\}$ ,  
waarbij  $n_0(x)$  het aantal nullen in  $x$  voorstelt

Gebruik het `egrep` commando om enkel die regels van het bestand `geheim.txt` te selecteren met een bitstring die tot een opgegeven verzameling behoort. Vermeld op je antwoordblad voor elke verzameling het gebruikte selectiecommando, en geef ook aan hoeveel regels je gevonden hebt.

2. De geheime boodschap bestaat uit drie woorden. Beschouw nu de verzamelingen  $A$ ,  $B$  en  $C$  zoals hierboven gedefinieerd. Het raadsel kan dan op de volgende manier worden opgelost:
  - (a) Het eerste woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap B$ .
  - (b) Het tweede woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $A \cap C$ .
  - (c) Het derde woord wordt gevonden op de unieke regel met de bitstring uit de verzameling  $B \cap C$ .

Vermeld op je antwoordblad voor elk gevonden woord het `egrep` commando (of de commandosequentie) die je gebruikt hebt om het woord te vinden. Geef ook de oplossing van het raadsel.

### Opgave 2

Gegeven is een bestand `films.txt`, waarvan de regels de volgende vorm hebben:

```
<titel> <jaar> <landcode> <IMDB> <score> <aantal> <classificatie>
```

Elke regel bevat informatie over een film, waarbij een spatie wordt gebruikt als veldscheidingsteken. Dit betekent dus dat er geen spaties meer voorkomen in de inhoud van de velden zelf. Het IMDB-veld verwijst naar een webpagina van de vorm `http://www.imdb.com/title/ttx/`, waarbij  $x$  een willekeurig aantal cijfers voorstelt. Een score is van de vorm  $x.y$  en staat op 10. Een score van 10/10 is niet mogelijk. De score wordt gevolgd door het aantal uitgebrachte stemmen voor de film.

Het classificatieveld is van de vorm `landcode: classificatie`. Verder bevat het bestand ook nog commentaarregels die beginnen met een hekje (`#`).

Gevraagd wordt om, gebruik makend van de teksteditor `vi`, een reeks substitutiecommando's te geven die achtereenvolgens de volgende opdrachten uitvoeren.

1. Kort het IMDB-veld in zodat enkel het titelnummer behouden blijft.  
Bijvoorbeeld, `http://www.imdb.com/title/tt0110912/` wordt `0110912`.
2. Vervang de films waarvan de opgegeven landcode gelijk is aan deze die deel uitmaakt van de classificatie door een lege lijn. "Gosford Park" zal zo bijvoorbeeld verwijderd worden.
3. Verander het classificatieformaat voor alle films die een minimum leeftijd hebben (m.a.w. classificaties van de vorm `land:code-leeftijd`) naar `leeftijd:land-code`.
4. Markeer de lijnen met een minimum score van 8/10 en minimum 10000 stemmen door achteraan een sterretje (`*`) te plaatsen.
5. Reorganiseer het bestand zodanig dat tussen elke filmregel een lege lijn komt te staan en de films alfabetisch gerangschikt zijn. Het kan zijn dat je hiervoor meer dan één commando nodig hebt.

Probeer zo weinig mogelijk commando's te geven en zo weinig mogelijk tekens te gebruiken in je commando's. De commentaarregels in `films.txt` mogen door je substituties niet gewijzigd worden. Alle wijzigingen moeten na elkaar uitgevoerd worden. Voeg de gebruikte substitutiecommando's toe aan je antwoordbestand.

### Opgave 3

Gegeven zijn de volgende tekstfragmenten:

1. **Lemma 1** *De formule*

$$\lim_{\delta \rightarrow +\infty} \left[ 1 + \left( \left( \frac{1-a}{a} \right)^\delta + \left( \frac{1-b}{b} \right)^\delta \right)^{\frac{1}{\delta}} \right]^{-1} = \min\{a, b\}$$

is geldig voor alle  $(a, b) \in [0, 1]^2$ .

**Stelling 2** *Voor alle  $a$  en  $b$  in  $[0, 1]$  geldt:*

$$\lim_{\delta \rightarrow +\infty} \left[ 1 + \left( \left( \frac{1-a}{a} \right)^\delta + \left( \frac{1-b}{b} \right)^\delta \right)^{\frac{1}{\delta}} \right]^{-1} = \begin{cases} a & \text{indien } a < b \\ b & \text{anders} \end{cases}$$

2. Zij  $U$  een universum,  $X \subseteq U$  en  $Y \subseteq U$ . Als  $\mu_X$  en  $\mu_Y$  afbeeldingen zijn van  $U$  naar  $\{0, 1\}$  zodanig dat

$$(\forall u \in U)(\mu_X(u) = 0 \iff u \notin X) \quad \text{en} \quad (\forall u \in U)(\mu_Y(u) = 0 \iff u \notin Y),$$

dan geldt: 
$$\frac{|A| + |B| - |A \cup B|}{|A \cup B|} = \frac{\sum_{u \in U} \mu_X(u) \cdot \mu_Y(u)}{\sum_{u \in U} (\mu_X(u) + \mu_Y(u) - \mu_X(u) \cdot \mu_Y(u))}.$$

3. Beschouw de  $[0, 1]^2 \rightarrow [0, 1]$  afbeeldingen  $S_{\mathbf{P}}$  en  $T_{\mathbf{P}}$  die gegeven worden door:

$$S_{\mathbf{P}}(a, b) = a + b - a \cdot b \quad (1)$$

$$T_{\mathbf{P}}(a, b) = 1 - S_{\mathbf{P}}(1 - a, 1 - b) \quad (2)$$

voor alle  $(a, b) \in [0, 1]^2$ . De gelijkheid  $T_{\mathbf{P}}(a, b) = a \cdot b$  geldt voor willekeurige  $a$  en  $b$  in  $[0, 1]$ , vermits:

$$\begin{aligned} T_{\mathbf{P}}(a, b) &= 1 - S_{\mathbf{P}}(1 - a, 1 - b) && \text{want (2)} \\ &= 1 - ((1 - a) + (1 - b) - (1 - a) \cdot (1 - b)) && \text{want (1)} \\ &= 1 - ((1 - a) + (1 - b) - (1 - a - b + a \cdot b)) \\ &= 1 - 1 + a - 1 + b + 1 - a - b + a \cdot b \\ &= a \cdot b \end{aligned}$$

4. De volgende tabel geeft een overzicht van de meest recente versie-nummers van het softwarepakket METAFONT, dat deel uitmaakt van T<sub>E</sub>X:

VERSIENUMMERS METAFONT:	GEBRUIKELIJKE NUMMERS:
2.718	2.3
2.7182	2.4
2.71828	2.5

Zoals blijkt uit deze tabel, naderen de versie-nummers van METAFONT naar  $e$ .

Gevraagd wordt om L<sup>A</sup>T<sub>E</sub>X-code te genereren die precies hetzelfde resultaat oplevert als de tekstfragmenten binnen de bovenstaande kaders. Zorg ervoor dat zowel de L<sup>A</sup>T<sub>E</sub>X-broncode die overeenstemt met elk tekstfragment als het eindresultaat zichtbaar zijn in de gecompileerde versie (PostScript of PDF) van je antwoordbestand.

## Opgave 4

Gegeven zijn de volgende bestanden: `jukebox.txt`, `lettersA.txt`, `lettersB.txt`, `lettersC.txt`, `jukebox_e.txt`, `substitute`, `alphabet.txt` en `keyfile.txt`.

1. (a) Implementeer de volgende (eenvoudige) commando's aan de hand van shell files:
  - i. `./filter`  
Verwijdert alle karakters die geen letter zijn uit standaard invoer en schrijft het resultaat naar standaard uitvoer. Met andere woorden, alles wat geen letter is wordt weggefilterd.
  - ii. `./rows`  
Zet alle karakters die binnenkomen via standaard invoer op een aparte regel, en stuurt het resultaat naar standaard uitvoer.
  - iii. `./unisort`  
Sorteert de regels die binnenkomen via standaard invoer alfabetisch, en verwijdert eventuele dubbele voorkomens van een regel. Het resultaat is dus een gesorteerde lijst van unieke regels. Dit resultaat wordt naar standaard uitvoer geschreven.
- (b) Combineer deze commando's vervolgens tot (telkens) één samengesteld commando dat:
  - i. het bestand `jukebox.txt` omzet naar `lettersA.txt`.
  - ii. het bestand `lettersB.txt` genereert, vertrekkende van `jukebox.txt`.
  - iii. het bestand `lettersC.txt` genereert, vertrekkende van `lettersA.txt`.

2. (a) Gebruik opnieuw shell files om de volgende (meer complexe) commando's te implementeren:

i. `./unify <bestand 1> <bestand 2> ... <bestand n>`

Plakt de overeenkomstige regels van standaard invoer en de opgegeven bestanden aan elkaar, gescheiden door een TAB, en stuurt het resultaat naar standaard uitvoer.

**Voorbeeld:** Stel dat de bestanden `2.txt` en `3.txt` beide slechts één regel bevatten, en dat die regels respectievelijk uit het karakter “2” en “3” bestaan (gevolgd door een newline). Het commando `./unify 2.txt 3.txt` zal dan de regel “2 3” (en daarna een newline) uitschrijven. Als dit commando bovendien de regel “1” ontvangt via standaard invoer, dan wordt niet “2 3” maar “1 2 3” uitgeschreven.

ii. `./createkey <bestand 1> <bestand 2> ... <bestand n>`

Stuurt een lijst van karaktersequenties naar standaard uitvoer, waarbij elke karaktersequentie overeenkomt met de inhoud van minstens één regel in één van de opgegeven bestanden. Als een karaktersequentie *a* vóór een andere karaktersequentie *b* staat in de teruggegeven lijst, dan wil dat zeggen dat *a* meer voorkomt dan *b*. Met andere woorden, het eerste karakter komt het meeste voor, en het laatste het minste.

**Voorbeeld:** Het gegeven bestand `keyfile.txt` bevat de uitvoer van het commando `./createkey lettersB.txt`.

(b) De shell file `substitute` krijg je van ons cadeau. Deze file implementeert het commando `./substitute <bestand>`, waarmee karakters in `<bestand>` kunnen vervangen worden. De regels die dit commando ontvangt via standaard invoer, moeten bestaan uit twee karakters gescheiden door een TAB. Elk voorkomen in `<bestand>` van het karakter voor de TAB zal dan telkens vervangen worden door het karakter erna. Het resultaat wordt uitgeschreven naar standaard uitvoer (en wordt dus niet opgeslagen in `<bestand>`).

Achterhaal wat het commando `./substitute` precies doet, en combineer het vervolgens met `./unify` en `./createkey` tot (telkens) één samengesteld commando dat:

i. Het bestand `jukebox.e.txt` genereert, vertrekkende van de bestanden `jukebox.txt`, `keyfile.txt` en `lettersC.txt`. Dit houdt in dat elk voorkomen in `jukebox.txt` van de eerste letter in `lettersC.txt` wordt vervangen door de letter die het meeste voorkomt in `jukebox.txt`, elk voorkomen van de tweede letter in `lettersC.txt` wordt vervangen door de op één na meest voorkomende letter, etc.

ii. Het bestand `jukebox.txt` genereert, vertrekkende van `jukebox.e.txt`, `lettersB.txt` en `lettersC.txt`.

Merk op dat `jukebox.e.txt` onleesbaar is, en dat het behoorlijk moeilijk is om het om te zetten naar `jukebox.txt` als je niet over `keyfile.txt` beschikt (en ook niet weet hoe je `keyfile.txt` moet genereren). Met andere woorden, de hierboven besproken commando's implementeren een (zeer primitieve) vorm van encryptie.

## Opgave 5

Veel gegevens bevatten een ruimtelijke component, waardoor we die gegevens een plaats op een landkaart kunnen geven. Het proces waarbij een object op een kaart wordt geplaatst, wordt *geocoderen* genoemd. Een *geocoder* is dan weer een stuk software of een webservice die helpt in dit proces. In deze opgave zullen we gebruik maken van zo een geocoder webservice voor het opvragen van geografische informatie die we op een website zullen plaatsen.

1. De broncode van de website <http://www.globalcomputing.com/states.html> bevat een lijst van alle staten van Amerika, en hun ISO 3166-2 staatcode die bestaat uit twee letters (de staatcode van Arizona is bijvoorbeeld AZ). Gevraagd wordt om op basis van deze lijst een shell script `stateCode` te schrijven, dat de Engelstalige naam van een Amerikaanse staat als argument

meekrijgt, en de corresponderende staatcode uitschrijft naar standaard uitvoer. Zorg ervoor dat het shell script niet hoofdlettergevoelig is wat betreft het doorgegeven argument. Hierbij is het toegelaten dat het shell script automatisch de broncode van de opgegeven website lokaal kopieert, op voorwaarde dat deze tijdelijke kopie bij het afsluiten van het shell script terug verwijderd wordt. Enkele voorbeelden ter controle:

```
./stateCode Texas      -->   TX
./stateCode Ohio      -->   OH
./stateCode Vermont   -->   VT
```

Let er dus op dat de staatcode niet altijd gelijk is aan de eerste twee letters van de naam van de staat.

2. Gebruik het shell script `stateCode` bij het laten genereren van een webpagina door een shell script `showInfo`. Dit shell script krijgt als argumenten de Engelstalige naam van een Amerikaanse stad en staat mee. Zo genereert het commando

```
./showInfo Phoenix Arizona
```

bijvoorbeeld een website die gebruik maakt van frames, en die er in een browser ongeveer uitziet zoals in de onderstaande figuur staat weergegeven.

The screenshot shows the Wikipedia page for Arizona. At the top, there is a yellow header with the text "Phoenix, Arizona (-112.07,33.44) (NL, FR, EN)". Below this is the Wikipedia logo and navigation tabs for "artikel", "overleg", "bewerk", and "geschiedenis". The main content area is titled "Arizona" and contains a table of contents with sections: 1 Geschiedenis, 2 Geografie, 2.1 Klimaat, 3 Demografie en economie, 4 Politiek, 5 Bestuurlijke indeling, 6 Zie ook, and 7 Externe links. The "Geschiedenis" section is expanded, showing text about the region's history. On the right side, there is a sidebar titled "State of Arizona" which includes the state flag, the state seal, the nickname "The Grand Canyon State", a map of the United States with Arizona highlighted in red, and a table with the following data:

Hoofdstad	Phoenix
Gouverneur	Janet Napolitano
Datum van toetreden	14 februari 1912

Het bovenste frame bevat de naam van de stad en de staat (gescheiden door een komma) die als argumenten aan het shell script werden doorgegeven, gevolgd door de coördinaten van de stad (breedte- en lengtegraden). Deze coördinaten kunnen worden opgevraagd via de webservice

```
http://api.local.yahoo.com/MapsService/V1/geocode?appid=ugent_cg&city=???&state=???
```

waarbij je op de plaats van de ??? respectievelijk de naam van een Amerikaanse stad en de code van de staat waarin die ligt meegeeft. Voor het voorbeeld van Phoenix, Arizona wordt dit

```
http://api.local.yahoo.com/MapsService/V1/geocode?appid=ugent_cg&city=Phoenix&state=AZ
```

Deze webservice geeft dan een XML-bestand terug, waaruit de coördinaten van de stad kunnen gefilterd worden. Voor ons voorbeeld ziet dit XML-bestand er als volgt uit:

```
<?xml version="1.0"?>
<ResultSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xmlns="urn:yahoo:maps" xsi:schemaLocation="urn:yahoo:maps
           http://api.local.yahoo.com/MapsService/V1/GeocodeResponse.xsd">
  <Result precision="city">
    <Latitude>33.447201</Latitude>
    <Longitude>-112.073166</Longitude>
    <Address></Address>
    <City>PHOENIX</City>
    <State>AZ</State>
    <Zip></Zip>
    <Country>US</Country>
  </Result>
</ResultSet>
```

Zorg ervoor dat op de webpagina enkel twee cijfers na de komma van de coördinaten worden weergegeven. Na de coördinaten bevat het bovenste frame ook nog drie links naar de Nederlandstalige (NL), Franstalige (FR) en Engelstalige (EN) wikipagina's van de staat die als parameter werd doorgegeven aan het shell script. Voor de staat Arizona zien de URLs van deze wikipagina's er respectievelijk als volgt uit:

```
href=http://nl.wikipedia.org/wiki/Arizona
href=http://fr.wikipedia.org/wiki/Arizona
href=http://en.wikipedia.org/wiki/Arizona
```

waarbij het laatste stuk van de URLs dus telkens bestaat uit de naam van de staat die als parameter aan het shell script werd doorgegeven. Bij het aanklikken van de hyperlinks (NL, FR, EN) moet de corresponderende webpagina in het onderste frame getoond worden. Zorg er ook voor dat de Nederlandstalige wikipagina van de opgegeven staat standaard getoond wordt in het onderste frame. Bovendien moeten alle webpagina's die het script genereert voldoen aan de XHTML 1.0 standaard (controleer dit aan de hand van de W3C validator). Kopieer voorbeelden van gegenereerde XHTML bestanden naar de USB-stick. Je kan het resultaat testen aan de hand van de volgende voorbeelden

```
./showInfo Austin Texas
./showInfo Chicago Illinois
./showInfo Detroit Michigan
./showInfo Columbus Ohio
```

**Hint:** Je kan backquotes gebruiken om commando-substitutie uit te voeren. Bij het commando

```
echo staatcode: './stateCode Arizona'
```

zal eerst het stuk tussen de backquotes worden uitgevoerd (`./stateCode Arizona`), en wordt daarna de uitvoer op de commandolijn gesubstitueerd. De commando-lijn wordt dus `echo staatcode: AZ`, waardoor dus de tekst `staatcode: AZ` naar standaard uitvoer wordt geschreven.

**Hint:** Witruimte wordt genegeerd in XML-bestanden. Je gaat er dus best niet van uit dat de verschillende XML-elementen op een nieuwe regel beginnen.