EXAMEN: **Computergebruik**

1e Bachelor Informatica — maandag 10-01-2011, 8:30
prof. dr. Peter Dawyndt — academiejaar 2010-2011
groep 1 — eerste zittijd

Opgave 1

Gegeven is een tekstbestand `film.txt` waarin de IMDB-rating van een aantal films werd opgenomen. Elke regel bevat de volgende informatie over een film: *i*) distributie van toegekende punten, *ii*) aantal uitgebrachte stemmen op de film, *iii*) berekende score en *iv*) titel gevolgd door jaar waarin film werd uitgebracht tussen ronde haakjes. De score is een getal uit het interval $[0, 10[$ dat steeds met één cijfer na de komma wordt weergegeven. De verschillende informatievelden worden van elkaar gescheiden door één of meerdere spaties, waarbij het laatste veld zelf ook spaties bevat. Het bestand bevat bovendien ook nog een hoofding waarvan alle regels beginnen met een hekje (`#`). Gevraagd wordt om, gebruik makend van de teksteditor `vi` (of `vim`), een reeks (substitutie)commando's op te stellen die achtereenvolgens de volgende opdrachten uitvoeren:

1. Zorg er voor dat de vier velden gescheiden worden door een puntkomma (`;`), waarbij ook alle spaties vooraan en achteraan elk veld verwijderd worden. Zo moeten de regels

   ```
   0000000125  540497   9.2  THE SHAWSHANK REDEMPTION (1994)
    0000000123  275216   8.7  THE USUAL SUSPECTS (1995)
   ```

   worden omgezet naar

   ```
   0000000125;540497;9.2;THE SHAWSHANK REDEMPTION (1994)
   0000000123;275216;8.7;THE USUAL SUSPECTS (1995)
   ```

2. Enkele regels komen meerdere keren in het bestand voor. Zorg er voor dat deze regels ontdubbeld worden. Sorteer de films ook volgens score van hoog naar laag. Als twee films een gelijke score hebben, dan moet de film met het meest aantal stemmen eerst komen.

3. Verander de volgorde van de informatievelden. Plaats de naam van de film vooraan, gevolgd door de score, het aantal stemmen en de distributie van de punten. Plaats dit laatste veld tussen ronde haakjes. De twee regels uit het voorgaande voorbeeld moeten dus omgezet worden naar

   ```
   THE SHAWSHANK REDEMPTION (1994);9.2;540497;(0000000125)
   THE USUAL SUSPECTS (1995);8.7;275216;(0000000123)
   ```

4. Plaats het jaartal van de film in een eigen veld. Dit levert het volgende resultaat op

   ```
   THE SHAWSHANK REDEMPTION;1994;9.2;540497;(0000000125)
   THE USUAL SUSPECTS;1995;8.7;275216;(0000000123)
   ```

5. Zorg dat elk woord van de filmtitels begint met een hoofdletter. Woorden worden gescheiden door spaties. Zo moeten de twee voorbeeldregels worden omgezet naar

   ```
   The Shawshank Redemption;1994;9.2;540497;(0000000125)
   The Usual Suspects;1995;8.7;275216;(0000000123)
   ```

   Hint: Als je werkt met een groep `\n` bij het vervangen, dan kan je deze groep laten voorafgaan door de modifiers `\U` en `\L` om alle karakters uit de groep bij de vervanging om te zetten naar hoofdletters (`\U\n`) of kleine letters (`\U\n`). Alternatief zorgen de modifiers `\u` en `\l` er voor

dat enkel het eerste karakter van de groep wordt omgezet naar een hoofdletter of kleine letter. Deze modifiers werken ook met het vervangsymbool `&`.

6. Plaats voor de titel van elke film met score groter of gelijk aan 9 een plusteken (`+`), voor elke film met score kleiner dan 4.5 een minteken (`-`) en een gelijkteken (`=`) voor alle andere films. Toegepast op de voorbeeldregels wordt dit

```
+The Shawshank Redemption;1994;9.2;540497;(0000000125)
=The Usual Suspects;1995;8.7;275216;(0000000123)
```

Probeer voor elke opdracht zo weinig mogelijk commando's te gebruiken en zorg er voor dat elk van deze commando's bestaat uit zo weinig mogelijk tekens. De regels van de hoofding mogen door je commando's niet gewijzigd worden, zelfs niet als er bijkomende regels aan de hoofding worden toegevoegd. Alle wijzigingen moeten na elkaar uitgevoerd worden.

Opgave 2

Gebruik filters, I/O redirection en pipes om telkens commando's samen te stellen die uitvoer genereren die voldoet aan onderstaande beschrijvingen. Hierbij is het toegelaten om gebruik te maken van `sed` of `gsed`, maar niet van andere programmeerbare filters zoals `awk`, `perl`, ... Vermijd dat je commando's (tijdelijke) bestanden aanmaken binnen het bestandssysteem, tenzij dat expliciet gevraagd wordt.

1. Gegeven is het commando

   ```
   for i in {1..42}; do echo $i $RANDOM; done
   ```

   dat de reeks gehele getallen van 1 tot en met 42 uitschrijft naar afzonderlijke regels op standaard uitvoer. Hierbij wordt elk van deze getallen telkens gevolgd door een spatie en een willekeurig geheel getal. Breid dit commando nu uit zodat het zes willekeurige lottogetallen in oplopende volgorde uitschrijft op één enkele regel, waarbij de getallen van elkaar worden gescheiden door een koppelteken (`-`). De uitvoer is dan bijvoorbeeld

   ```
   4-15-20-31-33-38
   ```

   Hierbij ga je als volgt te werk. Door de oorspronkelijk regels eerst te sorteren op basis van de gegenereerde willekeurige getallen (die je daarna niet meer nodig hebt), en daarvan enkel de eerste zes regels over te houden, krijg je een lijst van zes willekeurige getallen tussen 1 en 42. Rest dan enkel nog om de resterende regels zó te manipuleren dat alle getallen op één enkele regel komen te staan, gescheiden door koppeltekens.

2. Het bestand `beurs.txt` bevat informatie over de aandelenkoers van een aantal informaticabedrijven die op NASDAQ genoteerd staan. Elke regel begint met de naam van het bedrijf, gevolgd door de openingskoers en de slotkoers. Deze drie informatievelden worden van elkaar gescheiden door een dubbelpunt (`:`). De stand van een aandelenkoers wordt weergegeven met een komma (`,`) als decimaalteken. Zo bevat het bestand bijvoorbeeld de volgende regels

   ```
   Apple Inc.:325,64:329,57
   Microsoft Corporation:28,05:27,98
   ```

   Stel een commando op dat de inhoud van het bestand `beurs.txt` weergeeft op standaard uitvoer, waarbij elke regel achteraan wordt aangevuld met een extra veld dat de winst of het verlies in de aandelenkoers (slotkoers-openingskoers) voor die dag uitdrukt. Bovenstaande voorbeeldregels moeten dan bijvoorbeeld worden uitgeschreven als

   ```
   Apple Inc.:325,64:329,57:3,93
   Microsoft Corporation:28,05:27,98:-0,07
   ```

Let hierbij op het feit dat ook de waarde in het bijkomende veld moet worden weergegeven met een komma (`,`) als decimaalteken. Indien er enkel een nul voor het decimaalteken staat, dan mag deze niet weggelaten worden. Met andere woorden `0,03` mag bijvoorbeeld niet worden weergegeven als `,03`, en `-0,07` ook niet als `-,07`.

3. Veronderstel dat de directory `dir2` initieel een exacte kopie was van de directory `dir1`, maar dat achteraf nog enkele wijzigingen werden aangebracht aan sommige bestanden onder `dir2` (inhoud gewijzigd, bestanden verwijderd, bestanden toegevoegd). We willen nu nagaan welke bestanden gewijzigd werden. Hiervoor moet je zelf testdirectories `dir1` en `dir2` aanmaken en op basis daarvan de volgende vier commando's opstellen.

   (a) Het onderstaande commando illustreert hoe je een unieke sleutel (een *md5 checksum* in dit geval) kunt laten berekenen op basis van de inhoud van een bestand. Hierbij wordt `/usr/bin/perl` als voorbeeldbestand gebruikt.

   ```
   $ openssl dgst -md5 /usr/bin/perl
   MD5(/usr/bin/perl)= 2bfdff8cd9e6012a5ba3f95d653df476
   ```

   Gebruik deze informatie om een commando op te stellen dat voor elk gewoon bestand (*regular file*) dat zich onder de directory `dir1` of één van de subdirectories daaronder bevindt een regel uitschrijft, die de *basename* van het bestand weergeeft (dus de naam van het bestand zonder bijkomende directory-informatie) gevolgd door een spatie en de *md5 checksum* die berekend werd op basis van de inhoud van het bestand. Het commando moet deze regels vervolgens alfabetisch sorteren en wegschrijven naar het bestand `md5dir1.txt` in de huidige directory. De inhoud van dit bestand zou er als volgt kunnen uitzien

   ```
   test1 6ddb4095eb719e2a9f0a3f95677d24e0
   test2 59dd3f9ecdec2a5dc45c99b7b093f8bf
   ```

   (b) Geef een commando dat een analoog resultaat genereert als beschreven in het voorgaande puntje, maar dan voor de directory `dir2`, en die het resultaat wegschrijft naar het bestand `md5dir2.txt` in de huidige directory.

   (c) Geef een commando dat gebruik maakt van de bestanden `md5dir1.txt` en `md5dir2.txt` om een lijst van de bestanden die voorkomen onder directory `dir1` en die niet of met gewijzigde inhoud voorkomen onder `dir2` uit te schrijven naar standaard uitvoer. Merk op dat bestanden waarvan de inhoud ongewijzigd is, maar waarvan de relatieve locatie in `dir2` anders is dan in `dir1`, niet in deze lijst mogen voorkomen.

   (d) Geef een manier om de voorgaande commando's te bundelen tot één enkele commandolijn. Optimaal zoeken we een manier die vermijdt dat de tijdelijke bestanden `md5dir1.txt` en `md5dir2.txt` moeten aangemaakt worden, en die toch hetzelfde resultaat oplevert.

Opgave 3

Het genoom van een levend organisme bestaat uit een opeenvolging van genen. Deze genen coderen voor de eiwitten die het functioneren van de cellen sturen waaruit het organisme is opgebouwd. Genen kunnen voorwaarts of achterwaarts georiënteerd zijn en overlappen elkaar niet. Tussen twee genen kunnen fragmenten liggen die niet coderen voor eiwitten. Deze niet-coderende fragmenten worden soms ook *junk DNA* genoemd. Een genoomfragment wordt grafisch op de volgende manier voorgesteld:

- Het aantal symbolen dat gebruikt wordt voor de grafische voorstelling van een gen of een niet-coderend fragment staat in verhouding tot de lengte ervan op het genoom.
- Een voorwaarts gen wordt voorgesteld door nul of meer opeenvolgende gelijktekens (`=`) gevolgd door een *groter dan* teken (`>`). Zo wordt een voorwaarts gen van lengte vier voorgesteld door de tekenreeks `===>` en stelt de tekenreeks `>` een voorwaarts gen van lengte één voor.

- Een achterwaarts gen wordt voorgesteld door een *kleiner dan* teken (`<`) gevolgd door nul of meer opeenvolgende gelijktekens (`=`).
- Een niet-coderend fragment wordt voorgesteld door één of meer opeenvolgende koppeltekens (`-`).
- Wanneer een achterwaarts gen onmiddellijk gevolgd wordt door een voorwaarts gen — zonder tussenliggend niet-coderend fragment — en wanneer bovendien minstens één van beide genen minstens lengte twee heeft, dan worden deze genen in de grafische voorstelling van elkaar gescheiden door een verticale streep (`|`). Op die manier kan ondubbelzinnig bepaald worden welke gelijktekens bij welk gen behoren. De grafische voorstelling `<==>` kan immers dubbelzinnig geïnterpreteerd worden als `<|==>`, `<=|=>` of `<==|>`. De grafische voorstelling `<>` kan daarentegen niet dubbelzinnig geïnterpreteerd worden, en stelt een achterwaarts gen van lengte één voor, gevolgd door een voorwaarts gen van lengte één.

Zo is de tekenreeks `----==>--<===--<==|===>----` de grafische voorstelling van een genoomfragment met een voorwaarts gen, gevolgd door twee opeenvolgende achterwaartse genen, gevolgd door een voorwaarts gen. Alle genen worden hierbij gescheiden door niet-coderende fragmenten, behalve de laatste twee genen. Omdat deze laatste twee genen een voorbeeld vomen van het geval beschreven in het laatste puntje in bovenstaande lijst, moet er in de grafische voorstelling een verticale streep tussen beide genen geplaatst worden.
Elke regel van het bestand `genoom.txt` bevat de grafische voorstelling van een genoomfragment, gevolgd door een spatie en een woord dat enkel bestaat uit letters van het alfabet. Alle genoomfragmenten bevatten minstens twee genen. Gevraagd wordt:

1. Bepaal reguliere expressies voor elk van de onderstaande verzamelingen, waarbij $\mathcal{G}$ de verzameling voorstelt van grafische voorstellingen voor genoomfragmenten met minstens twee genen in het formaat dat hierboven werd beschreven. Probeer deze reguliere expressies zo kort mogelijk te houden.

   (a) $\alpha = \{g \in \mathcal{G} \mid \text{eerste gen van } g \text{ is voorwaarts en laatste gen van } g \text{ is achterwaarts}\}$
   voorbeeld: `--===><===<===-----<==---` $\in \alpha$, `<====|==>----<=---===>` $\notin \alpha$

   (b) $\beta = \{g \in \mathcal{G} \mid g \text{ heeft minstens zes opeenvolgende genen met dezelfde oriëntatie}\}$

   (c) $\gamma = \{g \in \mathcal{G} \mid \text{alle genen in } g \text{ hebben minimaal lengte 2}\}$
   voorbeeld: `--=><===---=><==` $\in \gamma$, `===><---<=|>=>` $\notin \gamma$

   (d) $\delta = \{g \in \mathcal{G} \mid \text{alle reeksen gelijktekens in } g \text{ zijn even lang}\}$
   voorbeelden: `>----===>===><===<` $\in \delta$, `===>---<===<==|===>` $\notin \delta$

   Gebruik een commando uit de `grep` familie om enkel die regels van het bestand `genoom.txt` te selecteren met grafische voorstellingen van genoomfragmenten die behoren tot de opgegeven verzameling. Vermeld in je antwoordbestand voor elke verzameling het gebruikte selectiecommando, en geef telkens ook aan hoeveel regels je gevonden hebt.

2. Beschouw de verzamelingen α, β, γ en δ zoals hierboven gedefinieerd. Gebruik nu deze verzamelingen om op de volgende manier een boodschap bestaande uit vier woorden te achterhalen:

   (a) het eerste woord staat op de unieke regel met een genoomfragment uit de verzameling $\alpha \cap \beta$

   (b) het tweede woord staat op de unieke regel met een genoomfragment uit de verzameling $\beta \cap \gamma$

   (c) het derde woord staat op de unieke regel met een genoomfragment uit de verzameling $\gamma \cap \delta$

   (d) het vierde woord staat op de unieke regel met een genoomfragment uit de verzameling $\delta \cap \alpha$

Vermeld in je antwoordbestand de gevonden woorden, samen met het unix commando (of de commandosequentie) dat je gebruikt hebt om elk van deze woorden te vinden. Elk commando of elke commandosequentie moet dus als resultaat één van de gezochte woorden naar standaard uitvoer schrijven (zonder het genoomfragment dat het woord voorafgaat).

Opgave 4

1. Geef LaTeX-code die een reconstructie maakt van onderstaande tabel, samen met de bijhorende tekst. Zorg er daarbij voor dat de opmaak zo getrouw mogelijk behouden blijft.

Onderstaande tabel geeft de relatieve fout van enkele eigenwaardeberekeningen voor het Collatz-probleem en het Paine-probleem, berekend aan de hand van de 6^{e}-orde CPM[2,2].

	Collatz ($n = 128$)		Paine ($n = 192$)	
k	E_k	rel. fout	E_k	rel. fout
0	70.2836836876	4.6(-13)	1.5198658211	3.0(-13)
10	47444.28579306	7.7(-11)	37.9644258619	5.3(-11)
20	182548.2030025	3.6(-10)	123.4977068009	1.9(-10)
30	405381.9379249	1.2(-9)	443.8529598352	4.2(-10)
40	715945.489746	4.6(-9)	963.9644462621	7.3(-10)
50	1114238.858465	3.2(-9)	1684.0120143379	1.1(-9)

Het aantal samplepunten wordt aangeduid als n en de notatie $a(-b)$ staat voor $a\,10^{-b}$.

2. Geef LaTeX-code die precies hetzelfde resultaat oplevert als het tekstfragment in onderstaand kader. Maak hierbij omgevingen aan voor `Stelling` en `Bewijs deel`. Zorg er voor dat formules en eigen omgevingen automatisch genummerd worden, en gebruik waar mogelijk verwijzingen naar deze nummeringen.

Stelling 1 ***(De m-schatter heeft een asymptotisch normale verdeling)***
Zij $\hat{\theta}_n$ de m-schatter voor θ_0 die gedefinieerd wordt door de vector $m(X_i, \theta)$ van functies van de data en de parameter θ en veronderstel dat aan de volgende voorwaarden voldaan is:

(a) consistentie: $\hat{\theta}_n \xrightarrow{p} \theta_0$

(b) de vector $m(X_i, \theta)$ van functies is continu afleidbaar in θ over een (compacte) omgeving $\mathcal{N}(\theta_0)$ van θ_0 (dus moet ook gelden dat $\theta_0 \in int(\Theta)$)

(c) $E\left[\frac{\partial m(X_1,\theta)}{\partial \theta^T}\right]$ is niet singulier met $\frac{\partial m(X_i,\theta)}{\partial \theta^T}$ de $p \times p$ matrix van partiële afgeleiden van de elementen van de vector $m(\cdot)$ naar θ.

(d) dominantie: er bestaat een functie $g(\cdot)$ van de data met $\left|\left|\frac{\partial m(X_i,\theta)}{\partial \theta^T}\right|\right| \leq g(X_i)$ voor alle $\theta \in \mathcal{N}(\theta_0)$ en $E[g(X_1)] < +\infty$

Dan geldt: $n^{1/2}(\hat{\theta}_n - \theta_0) \xrightarrow{d} N\left(0, \left[E\left\{\frac{\partial m(X_1,\theta_0)}{\partial \theta^T}\right\}\right]^{-1} Var\{m(X_1,\theta_0)\}\left[E\left\{\frac{\partial m(X_1,\theta_0)}{\partial \theta^T}\right\}\right]^{-1^T}\right)$.

Bewijs deel 1 *We starten door op te merken dat $E\left[\frac{\partial m(X_1,\theta)}{\partial \theta^T}\right]$ continu is in elke $\theta \in \mathcal{N}(\theta_0)$ en dat*

$$\sup_{\theta \in \mathcal{N}(\theta_0)} \left|\left| \frac{1}{n}\sum_{i=1}^{n} \frac{\partial m(X_i,\theta)}{\partial \theta^T} - E\left[\frac{\partial m(X_1,\theta)}{\partial \theta^T}\right] \right|\right| \xrightarrow{p} 0. \tag{1}$$

Na (1) tonen we aan dat $\hat{\theta}_n$ asymptotisch lineair is en gaan we op zoek naar de invloedsfunctie van deze schatter. Hiertoe passen we de middelwaardestelling toe:

$$0 = \sum_{i=1}^{n} m(X_i, \hat{\theta}_n) = \sum_{i=1}^{n} m(X_i, \theta_0) + \left\{\sum_{i=1}^{n} \frac{\partial m(X_i, \theta_n^*)}{\partial \theta^T}\right\}(\hat{\theta}_n - \theta_0),$$

Bewijs deel 2 *Omdat $\hat{\theta}_n \xrightarrow{p} \theta_0$ en θ_n^* een waarde is die tussen $\hat{\theta}_n$ en θ_0 ligt, zal ook $\theta_n^* \xrightarrow{p} \theta_0$ en samen met de continuïteit van $E\left[\frac{\partial m(X_1,\theta)}{\partial \theta^T}\right]$, de continue afbeeldingsstelling en de uniforme convergentie (1) volgt hieruit:*

$$\begin{aligned}
\left|\left| \frac{1}{n}\sum_{i=1}^{n} \frac{\partial m(X_i,\theta_n^*)}{\partial \theta^T} - E\left[\frac{\partial m(X_1,\theta_0)}{\partial \theta^T}\right] \right|\right| &\leq \left|\left| \frac{1}{n}\sum_{i=1}^{n} \frac{\partial m(X_i,\theta_n^*)}{\partial \theta^T} - E\left[\frac{\partial m(X_1,\theta_n^*)}{\partial \theta^T}\right] \right|\right| \\
&\quad + \left|\left| E\left[\frac{\partial m(X_1,\theta_n^*)}{\partial \theta^T}\right] - E\left[\frac{\partial m(X_1,\theta_0)}{\partial \theta^T}\right] \right|\right| \\
&\leq \sup_{\theta \in \mathcal{N}(\theta_0)} \left|\left| \frac{1}{n}\sum_{i=1}^{n} \frac{\partial m(X_i,\theta)}{\partial \theta^T} - E\left[\frac{\partial m(X_1,\theta)}{\partial \theta^T}\right] \right|\right| \\
&\quad + \left|\left| E\left[\frac{\partial m(X_1,\theta_n^*)}{\partial \theta^T}\right] - E\left[\frac{\partial m(X_1,\theta_0)}{\partial \theta^T}\right] \right|\right| \\
&\xrightarrow{p} 0.
\end{aligned}$$

Plaats een PDF bestand met daarin de gecompileerde LaTeX fragmenten in het ZIP-bestand dat je indient via Indianio.

EXAMEN: **Computergebruik**

1e Bachelor Informatica — maandag 10-01-2011, 14:00
prof. dr. Peter Dawyndt — academiejaar 2010-2011
groep 2 — eerste zittijd

Opgave 1

Gegeven is een tekstbestand `NASDAQ.txt` waarin de beursnotering van aandelen op de NASDAQ werd opgenomen. Elke regel bevat de volgende informatie over een aandeel: *i*) volledige bedrijfsnaam, *ii*) afgekorte bedrijfsnaam, *iii*) datum waarop aandelenkoers betrekking heeft, *iv*) openingskoers, *v*) hoogste stand van koers die dag, *vi*) laagste stand van koers die dag, *vii*) slotkoers en *viii*) aantal verhandelde aandelen tijdens die dag. De verschillende informatievelden worden van elkaar gescheiden door een *komma* (`,`). Gevraagd wordt om, gebruik makend van de teksteditor `vi` (of `vim`), een reeks (substitutie)commando's op te stellen die achtereenvolgens de volgende opdrachten uitvoeren:

1. Zorg er voor dat de velden gescheiden worden door een asterisk (`*`), waarbij ook alle spaties vooraan en achteraan elk veld verwijderd worden. Zo moeten de regels

   ```
   Australia Acquisition Corp.,AAC,20110103 ,9.7,9.7,9.63,9.63,1100
    Apple Inc.,AAPL,20101207,325.64,330.26,324.84,329.57,15897200
   ```

   omgezet worden naar

   ```
   Australia Acquisition Corp.*AAC*20110103*9.7*9.7*9.63*9.63*1100
   Apple Inc.*AAPL*20101207*325.64*330.26*324.84*329.57*15897200
   ```

2. Sorteer de aandelen in dalende volgorde volgens het aantal dat ervan verhandeld werd. Indien aandelen met eenzelfde volume verhandeld werden, dan moeten deze aandelen alfabetisch gerangschikt worden volgens hun afgekorte bedrijfsnaam.

3. Bij de weergave van de stand van een beurskoers wordt gebruik gemaakt van een punt (`.`) als decimaalteken. Zorg er voor dat alle punten die als decimaalteken gebruikt worden door een komma (`,`) vervangen worden. Je mag er hierbij van uitgaan dat een punt als decimaalteken gebruikt wordt, als het wordt voorafgegaan en wordt gevolgd door een cijfer. Alle punten die niet als decimaalteken gebruikt worden moeten ongewijzigd blijven. De twee regels uit het voorgaande voorbeeld moeten dus omgezet worden naar

   ```
   Australia Acquisition Corp.*AAC*20110103*9,7*9,7*9,63*9,63*1100
   Apple Inc.*AAPL*20101207*325,64*330,26*324,84*329,57*15897200
   ```

4. Voeg punten (`.`) toe die de duizendtallen groeperen in groepjes van drie, in de weergave van het aantal verhandelde aandelen. Je mag er hierbij van uitgaan dat er van een bedrijf nooit meer dan 999.999.999 aandelen op één dag verhandeld worden. Dit levert het volgende resultaat op

   ```
   Australia Acquisition Corp.*AAC*20110103*9,7*9,7*9,63*9,63*1.100
   Apple Inc.*AAPL*20101207*325,64*330,26*324,84*329,57*15.897.200
   ```

5. De datum waarop de aandelenkoers betrekking heeft, wordt initieel weergegeven als een getal van 8 cijfers: 4 voor het jaar, 2 voor de maand en 2 voor de dag (`jjjjmmdd`). Pas deze voorstelling aan zodat de dag eerst komt, gevolgd door de maand en het jaar, telkens gescheiden door een slash (`/`). Geef hierbij enkel de laatste 2 cijfers van het jaar weer en laat bij de dag en de maand een eventuele voorloopnul (`0`) aan het begin weg. Zo moeten de twee voorbeeldregels worden

omgezet naar

```
Australia Acquisition Corp.*AAC*3/1/11*9,7*9,7*9,63*9,63*1.100
Apple Inc.*AAPL*7/12/10*325,64*330,26*324,84*329,57*15.897.200
```

6. Plaats vooraan elke regel met een aandeel dat minstens 1.000.000 keer werd verhandeld een *groter dan* teken (`>`), en vooraan elke regel met een aandeel dat minder dan 1.000 keer werd verhandeld een *kleiner dan* teken (`<`). Aan het begin van de andere regels met aandelen plaats je een tilde (`~`). Toegepast op de voorbeeldregels wordt dit

   ```
   ~Australia Acquisition Corp.*AAC*3/1/11*9,7*9,7*9,63*9,63*1.100
   >Apple Inc.*AAPL*7/12/10*325,64*330,26*324,84*329,57*15.897.200
   ```

Probeer voor elke opdracht zo weinig mogelijk commando's te gebruiken en zorg er voor dat elk van deze commando's bestaat uit zo weinig mogelijk tekens. Alle wijzigingen moeten na elkaar uitgevoerd worden.

Opgave 2

Gebruik filters, I/O redirection en pipes om telkens commando's samen te stellen die uitvoer genereren die voldoet aan onderstaande beschrijvingen. Hierbij is het toegelaten om gebruik te maken van `sed` of `gsed`, maar niet van andere programmeerbare filters zoals `awk`, `perl`, ... Vermijd dat je commando's (tijdelijke) bestanden aanmaken binnen het bestandssysteem, tenzij dat expliciet gevraagd wordt.

1. Geef een commando dat een overzicht naar standaard uitvoer schrijft van de interpreters die gebruikt worden in de *hashbang* (*shebang*) regel van de gewone bestanden (*regular files*) die zich in de directory `/usr/bin` of één van de subdirectories daaronder bevinden. De opties die eventueel worden meegegeven aan de interpreter op de hashbang regel moeten hierbij genegeerd worden. Zorg er voor dat in het overzicht enkel de 5 meest voorkomende interpreters weergegeven worden, volgens dalend aantal voorkomens. Op `genix` moet het commando bijvoorbeeld het volgende overzicht genereren.

   ```
   43 /bin/sh
   13 /usr/bin/perl
   12 /bin/ksh
   11 /usr/bin/sh
    7 /bin/bash
   ```

   Zorg er voor dat het commando in geen enkel geval andere informatie uitschrijft dan het gevraagde overzicht.

2. Elke regel van het bestand `BMI.txt` bevat de naam van een beroemdheid, gevolgd door haar lengte (in voet en inch; formaat *`voet' inch"`*) en haar gewicht in pond. Deze drie informatievelden worden van elkaar gescheiden door een dubbelpunt (`:`) en alle opgegeven waarden zijn gehele getallen. Zo bevat het bestand bijvoorbeeld de volgende regels

   ```
   Aniston, Jennifer:5' 5":112
   Banks, Tyra:5' 11":127
   ```

   Stel een commando op dat elke regel van het bestand `BMI.txt` omvormt tot een regel die de naam van de beroemdheid bevat, gevolgd door haar gewicht uitgedrukt in kilogram, haar lengte uitgedrukt in meter en haar *body mass index* (BMI). Deze vier informatievelden moeten opnieuw van elkaar gescheiden worden door een dubbelpunt (`:`). Zorg er voor dat alle reële waarden worden weergegeven met maximaal twee cijfers na de komma, en dat de regels volgens dalende

BMI-waarde naar standaard uitvoer worden uitgeschreven. De eerste regels van de uitvoer moeten er bijgevolg als volgt uitzien

```
Witherspoon, Reese:55.33:1.57:22.31
Spears, Britney:58.51:1.65:21.46
Barrymore, Drew:53.52:1.62:20.25
Winslet, Kate:58.96:1.72:19.76
...
```

Een lengte uitgedrukt in voet (v) en inch (i) kan worden omgezet naar een lengte uitgedrukt in meter volgens de formule $0.0254\,(12v + i)$. Een gewicht uitgedrukt in pond kan worden omgezet naar een gewicht uitgedrukt in kilogram, door de waarde te delen door 2.20462262185. De *body mass index* van iemand met een gewicht g (in kilogram) en lengte l (in meter) wordt berekend als $\frac{g}{l^2}$.

Het gevraagde commando moet gebruik maken van het commando `bc` voor het uitvoeren van de berekeningen. Let hierbij op het feit dat de parameter `scale` kan gebruikt worden om de precisie van de berekeningen te bepalen. Gebruik voor jouw berekeningen precisie `scale=10`. Het effect van de precisie wordt geïllustreerd door onderstaand voorbeeld

```
$ echo "1/3;scale=2;1/3" | bc
0
.33
```

3. Bij deze vraag moeten de volgende vier commando's opgesteld worden

   (a) Het speciale bestand `/dev/urandom` is een *non-blocking pseudo-random generator* die een reeks willekeurige bytes genereert, telkens als er uit gelezen wordt (zie `man urandom`). Het commando `strings` kan gebruikt worden om leesbare tekst te extraheren uit binaire bestanden (zie `man strings`). Gebruik deze informatie om een commando op te stellen die 10.000 regels met willekeurige binaire bytes genereert, daaruit de leesbare tekst extraheert, omzet naar kleine letters en alfabetisch gerangschikt wegschrijft naar het bestand `willekeurige_tekst.txt` in de huidige directory.

   (b) Geef een tweede commando dat het tekstbestand met daarin een lijst van engelse woorden afhaalt vanaf de URL `http://users.ugent.be/~pdawyndt/woordenboek.txt` en rechtstreeks opslaat onder de naam `engelse_woorden.txt` in de `tmp` directory onder je *home directory* (maak eerst een directory `tmp` aan onder je home directory indien deze nog niet bestaat). Bij voorkeur mag dit commando dus geen extra bestand genereren dat tijdelijk een andere naam of een andere locatie heeft. Zorg er voor dat de regels gesorteerd worden, vooraleer ze worden opgeslagen in een lokaal bestand.

   (c) Geef een derde commando dat enkel die woorden uit het bestand `willekeurige_tekst.txt` teruggeeft die ook voorkomen in het bestand `engelse_woorden.txt`.

   (d) Geef een manier om de voorgaande commando's te bundelen tot één enkele commandolijn. Optimaal zoeken we een manier die vermijdt dat de bestanden `engelse_woorden.txt` en `willekeurige_tekst.txt` tijdelijke moeten aangemaakt worden, en die toch hetzelfde resultaat oplevert.

Opgave 3

Het genoom van een levend organisme bestaat uit een opeenvolging van genen. Deze genen coderen voor de eiwitten die het functioneren van de cellen sturen waaruit het organisme is opgebouwd. Genen kunnen voorwaarts of achterwaarts georiënteerd zijn en overlappen elkaar niet. Tussen twee genen

kunnen fragmenten liggen die niet coderen voor eiwitten. Deze niet-coderende fragmenten worden soms ook *junk DNA* genoemd. Een genoomfragment wordt grafisch op de volgende manier voorgesteld:

- Het aantal symbolen dat gebruikt wordt voor de grafische voorstelling van een gen of een niet-coderend fragment staat in verhouding tot de lengte ervan op het genoom.
- Een voorwaarts gen wordt voorgesteld door nul of meer opeenvolgende gelijktekens (`=`) gevolgd door een *groter dan* teken (`>`). Zo wordt een voorwaarts gen van lengte vier voorgesteld door de tekenreeks `===>` en stelt de tekenreeks `>` een voorwaarts gen van lengte één voor.
- Een achterwaarts gen wordt voorgesteld door een *kleiner dan* teken (`<`) gevolgd door nul of meer opeenvolgende gelijktekens (`=`).
- Een niet-coderend fragment wordt voorgesteld door één of meer opeenvolgende koppeltekens (`-`).
- Wanneer een achterwaarts gen onmiddellijk gevolgd wordt door een voorwaarts gen — zonder tussenliggend niet-coderend fragment — en wanneer bovendien minstens één van beide genen minstens lengte twee heeft, dan worden deze genen in de grafische voorstelling van elkaar gescheiden door een verticale streep (`|`). Op die manier kan ondubbelzinnig bepaald worden welke gelijktekens bij welk gen behoren. De grafische voorstelling `<==>` kan immers dubbelzinnig geïnterpreteerd worden als `<|==>`, `<=|=>` of `<==|>`. De grafische voorstelling `<>` kan daarentegen niet dubbelzinnig geïnterpreteerd worden, en stelt een achterwaarts gen van lengte één voor, gevolgd door een voorwaarts gen van lengte één.

Zo is de tekenreeks `----==>--<===--<==|===>----` de grafische voorstelling van een genoomfragment met een voorwaarts gen, gevolgd door twee opeenvolgende achterwaartse genen, gevolgd door een voorwaarts gen. Alle genen worden hierbij gescheiden door niet-coderende fragmenten, behalve de laatste twee genen. Omdat deze laatste twee genen een voorbeeld vomen van het geval beschreven in het laatste puntje in bovenstaande lijst, moet er in de grafische voorstelling een verticale streep tussen beide genen geplaatst worden.
Elke regel van het bestand `genoom.txt` bevat de grafische voorstelling van een genoomfragment, gevolgd door een spatie en een woord dat enkel bestaat uit letters van het alfabet. Alle genoomfragmenten bevatten minstens twee genen. Gevraagd wordt:

1. Bepaal reguliere expressies voor elk van de onderstaande verzamelingen, waarbij $\mathcal{G}$ de verzameling voorstelt van grafische voorstellingen voor genoomfragmenten met minstens twee genen in het formaat dat hierboven werd beschreven. Probeer deze reguliere expressies zo kort mogelijk te houden.

   (a) $\alpha = \{g \in \mathcal{G} \mid \text{eerste en laatste gen van } g \text{ hebben dezelfde oriëntatie}\}$
   voorbeeld: `--===><===<==========><==>---` $\in \alpha$, `<====|==>----<=---===>` $\notin \alpha$

   (b) $\beta = \{g \in \mathcal{G} \mid g \text{ bevat geen opeenvolgende genen met dezelfde oriëntatie}\}$
   voorbeeld: `<====|==>----<=---===>` $\in \beta$, `--===><===<=========><==>---` $\notin \beta$

   (c) $\gamma = \{g \in \mathcal{G} \mid \text{alle genen in } g \text{ worden steeds gescheiden door een niet-coderend fragment}\}$
   voorbeelden: `====>-----<====-<==---<==` $\in \gamma$, `----====>-<|====>-=>><=<====` $\notin \gamma$

   (d) $\delta = \{g \in \mathcal{G} \mid \text{opeenvolgende genen met tegengestelde oriëntatie worden in } g \text{ steeds gescheiden door een niet-coderend fragment met lengte minstens drie}\}$
   voorbeeld: `--<===----=>` $\in \delta$, `--====>--<=` $\notin \delta$

   Gebruik een commando uit de `grep` familie om enkel die regels van het bestand `genoom.txt` te selecteren met grafische voorstellingen van genoomfragmenten die behoren tot de opgegeven verzameling. Vermeld in je antwoordbestand voor elke verzameling het gebruikte selectiecommando, en geef telkens ook aan hoeveel regels je gevonden hebt.

2. Beschouw de verzamelingen α, β, γ en δ zoals hierboven gedefinieerd. Gebruik nu deze verzamelingen om op de volgende manier een boodschap bestaande uit vier woorden te achterhalen:

   (a) het eerste woord staat op de unieke regel met een genoomfragment uit de verzameling $\alpha \cap \beta$

   (b) het tweede woord staat op de unieke regel met een genoomfragment uit de verzameling $\beta \cap \gamma$

   (c) het derde woord staat op de unieke regel met een genoomfragment uit de verzameling $\gamma \cap \delta$

   (d) het vierde woord staat op de unieke regel met een genoomfragment uit de verzameling $\delta \cap \alpha$

   Vermeld in je antwoordbestand de gevonden woorden, samen met het unix commando (of de commandosequentie) dat je gebruikt hebt om elk van deze woorden te vinden. Elk commando of elke commandosequentie moet dus als resultaat één van de gezochte woorden naar standaard uitvoer schrijven (zonder het genoomfragment dat het woord voorafgaat).

Opgave 4

1. Geef LaTeX-code die een reconstructie maakt van onderstaande tabel, samen met de bijhorende tekst. Zorg er daarbij voor dat de opmaak zo getrouw mogelijk behouden blijft.

Onderstaande tabel geeft de relatieve fout van enkele eigenwaardeberekeningen voor het Collatz-probleem en het Paine-probleem, berekend aan de hand van de 6^{e}-orde CPM[2,2].

	Collatz ($n = 128$)		Paine ($n = 192$)	
k	E_k	rel. fout	E_k	rel. fout
0	70.1836836876	4.6(-13)	1.5198658211	3.0(-13)
10	47444.18579306	7.7(-11)	37.9644258619	5.3(-11)
20	182548.2030025	3.6(-10)	123.4977068009	1.9(-10)
30	405381.9379249	1.2(-9)	443.8529598352	4.2(-10)
40	715945.489746	4.6(-9)	963.9644462621	7.3(-10)
50	1114238.858465	3.2(-9)	1684.0120143379	1.1(-9)

Het aantal samplepunten wordt aangeduid als n en de notatie $a(-b)$ staat voor $a\,10^{-b}$.

2. Geef LaTeX-code die precies hetzelfde resultaat oplevert als het tekstfragment in onderstaand kader. Maak hierbij omgevingen aan voor `Hulpstelling` en `Bewijs deel`. Zorg er voor dat formules en eigen omgevingen automatisch genummerd worden, en gebruik waar mogelijk verwijzingen naar deze nummeringen.

Hulpstelling 1 ***(Een continue afbeeldingsstelling voor uniforme convergentie)*** *Zij $X(\theta)$, $X_1(\theta)$, $X_2(\theta), \ldots$ een oneindige rij stochastische veranderlijken over een kansruimte $(\Omega, \mathcal{F}, \mathcal{P})$, die ook functie zijn van de parameter θ (functies van $\Omega \times \Theta$ in $\mathbb{R}^m$ of $\mathbb{R}^{m\times p}$) waarvoor $X_n \overset{p}{\rightrightarrows} X$ en zij $g(\cdot)$ een continue functie (van $\mathbb{R}^m$ of $\mathbb{R}^{m\times p}$ in $\mathbb{R}^q$ of $\mathbb{R}^{r\times s}$, $m, p, q, r, s \in \mathbb{N}^0$). Dan convergeert $g(X_n(\cdot))$ ook uniform in kans naar $g(X(\cdot))$.*

Bewijs deel 1 *Neem willekeurig $\epsilon > 0$. Voor elke $\delta > 0$ definiëren we de verzameling $\mathcal{B}_\delta$ van functies $x(\cdot)$ van Θ in $\mathbb{R}^m$ of $\mathbb{R}^{m\times p}$ als*

$$\mathcal{B}_\delta = \{x(\cdot) | (\exists y(\cdot), \theta \in \Theta)(||x(\theta) - y(\theta)|| \leq \delta \text{ en } ||g(x(\theta)) - g(y(\theta))|| > \epsilon)\}. \tag{1}$$

Uit (1) volgt: wanneer er een $\theta' \in \Theta$ bestaat waarvoor $||g(X(\theta')) - g(X_n(\theta'))|| > \epsilon$, dan geldt

- *ofwel $X(\cdot) \in \mathcal{B}_\delta$,*
- *ofwel $X(\cdot) \notin \mathcal{B}_\delta$,*

$$\text{d.w.z. } (\forall y(\cdot) \text{ en } \forall \theta \in \Theta)(||x(\theta) - y(\theta)|| \leq \delta \Rightarrow ||g(x(\theta)) - g(y(\theta))|| \leq \epsilon).$$

$$\rightarrow \text{in dit geval is dus } ||X(\theta') - X_n(\theta')|| > \delta.$$

Bewijs deel 2 *We kunnen dus het volgende schrijven:*

$$\begin{aligned}
& \mathcal{P}\left(\sup_{\theta\in\Theta} ||g(X(\theta)) - g(X_n(\theta))|| > \epsilon\right) \\
= \; & \mathcal{P}\left(\sup_{\theta\in\Theta} ||g(X(\theta)) - g(X_n(\theta))|| > \epsilon, X(\cdot) \in \mathcal{B}_\delta\right) \\
& \quad + \; \mathcal{P}\left(\sup_{\theta\in\Theta} ||g(X(\theta)) - g(X_n(\theta))|| > \epsilon, X(\cdot) \notin \mathcal{B}_\delta\right) \\
\leq \; & \mathcal{P}\left(X(\cdot) \in \mathcal{B}_\delta\right) \quad + \quad \mathcal{P}\left(\sup_{\theta\in\Theta} ||X(\theta) - X_n(\theta)|| > \delta\right)
\end{aligned}$$

De eerste term in het rechterlid van bovenstaande ongelijkheid convergeert naar 0 wanneer $\delta \to 0$, want $\lim_{\delta\to 0} \mathcal{B}_\delta = \emptyset$ wegens de continuïteit van g.

Plaats een PDF bestand met daarin de gecompileerde LaTeX fragmenten in het ZIP-bestand dat je indient via Indianio.