
EXAMEN: Computergebruik

1^e Bachelor Informatica
prof. dr. Peter Dawyndt
groep 1

woensdag 09-01-2013, 8:30
academiejaar 2012-2013
eerste zittijd

Opgave 1

(12 pt)

Gebruik filters, I/O redirection en pipes om telkens een commando samen te stellen dat uitvoer genereert conform onderstaande beschrijvingen. Hierbij is het toegelaten om gebruik te maken van **sed**, maar niet van andere programmeerbare filters zoals **awk**, **perl**, Vermijd dat de commando's (tijdelijke) bestanden aanmaken binnen het bestandssysteem, tenzij dat expliciet gevraagd wordt.

1. Iemand heeft een regel tekst getypt op onderstaand toetsenbord, maar heeft daarbij zijn handen één positie naar rechts verplaatst. Dat betekent bijvoorbeeld dat van de bovenste rij (met shift) niet de toetsenreeks **QWERTYUIOP** gebruikt werd, maar in plaats daarvan **WERTYUIOP{**, of op de onderste regel (zonder shift) **xcvbnm**, (inclusief de komma) in plaats van **zxcvbnm**.



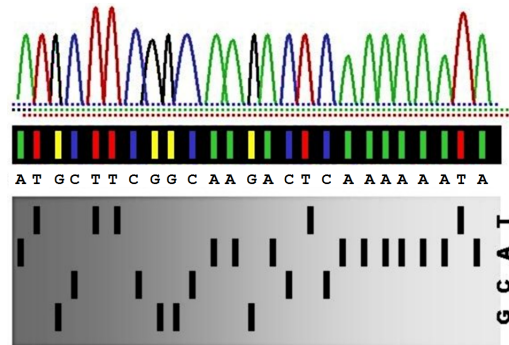
Vul onderstaande commando's op dezelfde manier aan, zodat de foutief ingetypte tekst gecorrigeerd wordt. Let hierbij op het feit dat het gebruik van hoofdletters en kleine letters moet gerespecteerd blijven.

```
$ echo "Jr;;p ept;f" | ...  
Hello world  
$ echo "Lmiyj od ,u jrtp" | ...  
Knuth is my hero
```

2. Geef een commando dat naar standaard uitvoer het totaal aantal pagina's uitschrijft van alle PDF bestanden (bestandsextensie **.pdf**) die zich onder de directory **/usr/share/doc** bevinden (inclusief de subdirectories) en die de tekst **game** in hun bestandsnaam hebben. Maak hierbij gebruik van onderstaand commando, dat onder andere het aantal pagina's van een gegeven PDF bestand uitschrijft.

```
$ pdftinfo -meta /usr/share/doc/texlive-games/latex/hexgame/hexgame.pdf  
Title:          hexgame.dvi  
Creator:        dvips(k) 5.95b Copyright 2005 Radical Eye Software  
Producer:       GPL Ghostscript 8.01  
CreationDate:   Thu Sep 20 20:47:09 2007  
ModDate:       Thu Sep 20 20:47:09 2007  
Tagged:         no  
Pages:          2  
Encrypted:      no  
Page size:     595 x 842 pts (A4)  
File size:     33071 bytes  
Optimized:     no  
PDF version:   1.2
```

3. Sanger sequencing is een methode die gebruikt wordt om het DNA van een organisme te bepalen. De methode werd in 1977 ontwikkeld door Frederick Sanger en zijn collega's, en was gedurende 25 jaar de meest gebruikte sequenceringsmethode. Hierbij worden de vier basen waaruit het DNA is opgebouwd (voorgesteld door de letters A, C, G en T) radioactief of fluorescent gelabeld, zodat opeenvolgende basen achtereenvolgens kunnen uitgelezen worden. Dit uitlezen gebeurt aan de hand van vier afzonderlijke probes, die voor elke positie nagaan of er een radioactief of fluorescent signaal gemeten wordt voor één van de vier mogelijke basen. Daarna worden de vier signalen samengevoegd tot één enkel signaal, waaruit de eigenlijke sequentie kan samengesteld worden.



Gegeven zijn vier tekstbestanden `sanger.a`, `sanger.c`, `sanger.g` en `sanger.t` die elk één regel tekst bevatten met het signaal voor één van de probes. Als de probe op een bepaalde positie een signaal heeft opgepikt, dan wordt dat voorgesteld met de bijhorende letter. Als er geen signaal werd opgepikt, dan wordt dit voorgesteld met een koppelteken. Geef een commando dat de inhoud van deze vier bestanden omzet naar de eigenlijke sequentie, die door het commando naar het bestand `sanger.seq` wordt uitgeschreven op één enkele regel. Het commando mag geen tijdelijke bestanden aanmaken. Indien dit niet lukt (of als vertrekbasis), zoek dan een oplossing die bestaat uit meerdere commando's en/of die tijdelijke bestanden aanmaakt (maar zorg er dan voor dat deze tijdelijke bestanden finaal terug verwijderd worden).

```
sanger.a      A----A-----A----A-A--AAAA-
sanger.c      -CC----CCCC-C-----CCC-C-----C
sanger.g      -----G---G--G--G-----G-----
sanger.t      ---TT-----T-T-----T-----
               =====
sanger.seq    ACCTTTAGCCCGCTGTAGCCACAGTAAAC
```

4. Voor deze opgave werken we met tekstbestanden die een reeks natuurlijke getallen bevatten, elk op een afzonderlijke regel. Geef één enkel commando dat, als het in een bash shell script `deelbaar` geplaatst wordt, aangeeft hoeveel getallen uit het bestand (het eerste getal buiten beschouwing gelaten) deelbaar zijn door het eerste getal uit het bestand. De locatie van een tekstbestand met natuurlijke getallen moet als (enige) argument aan het shell script meegegeven worden. Onderstaand voorbeeld geeft aan dat het bestand `getallen.txt` vier getallen bevat die deelbaar zijn door drie.

```
$ cat getallen.txt
3
1
51
966369
7
9
999996
11
$ deelbaar getallen.txt
4
```

Opgave 2

(12 pt)

In `straten.txt` kan je de informatie van alle straten van Gent in JSON-formaat vinden. Dit bestand bevat slechts 1 regel, die begint met `{"StratenInGent": [` gevolgd door een lijst van straatobjecten gescheiden door een komma, om te eindigen op `]`. Een straatobject begint met een `{`, eindigt met een `}` en bevat een aantal informatievelden gescheiden door een komma. Elk informatieveld heeft het formaat `"naam veld": "waarde veld"`. De volgorde van de informatievelden is: `postcode`, `straatcode`, `straatnaam`, `onpaar_van`, `onpaar_tot`, `paar_van`, `paar_tot`, `sector`, `stadsdeel`, `wijkNr`, `wijknaam`. 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. 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. Ter controle kan je gebruik maken van de meegeleverde bestanden `straten.i` ($1 \leq i \leq 6$), die telkens de inhoud van het bestand bevatten nadat de i -de opdracht werd uitgevoerd.

1. Zorg er voor dat elk straatobject op zijn eigen lijn komt te staan. De `{"StratenInGent": [` in het begin van het bestand en de `]` op het einde moeten weggelaten worden, net als de komma's tussen de straatobjecten. Bijvoorbeeld:

```
1 | {"StratenInGent": [{"postcode": "9000", "straatcode": "2101", "straatnaam": "Aaigemstraat", "onpaar_van": "", "onpaar_tot": "", "paar_van": "", "paar_tot": "", "sector": "A40", "stadsdeel": "Gent-Zuid", "wijkNr": "22", "wijknaam": "Stationsbuurt Noord"}, {"postcode": "9000", "straatcode": "7131", "straatnaam": "Banierstraat", "onpaar_van": "", "onpaar_tot": "", "paar_van": "2", "paar_tot": "30", "sector": "E32", "stadsdeel": "Gent-Noordoost", "wijkNr": "13", "wijknaam": "Dampoort"}, {"postcode": "9051", "straatcode": "8511", "straatnaam": "Broekkantstraat", "onpaar_van": "1", "onpaar_tot": "25", "paar_van": "2", "paar_tot": "60", "sector": "J200", "stadsdeel": "Gent-Zuid", "wijkNr": "27", "wijknaam": "Sint-Denijs-Westrem"}]}
```

wordt omgezet naar

```
1 | {"postcode": "9000", "straatcode": "2101", "straatnaam": "Aaigemstraat", "onpaar_van": "", "onpaar_tot": "", "paar_van": "", "paar_tot": "", "sector": "A40", "stadsdeel": "Gent-Zuid", "wijkNr": "22", "wijknaam": "Stationsbuurt Noord"}
2 | {"postcode": "9000", "straatcode": "7131", "straatnaam": "Banierstraat", "onpaar_van": "", "onpaar_tot": "", "paar_van": "2", "paar_tot": "30", "sector": "E32", "stadsdeel": "Gent-Noordoost", "wijkNr": "13", "wijknaam": "Dampoort"}
3 | {"postcode": "9051", "straatcode": "8511", "straatnaam": "Broekkantstraat", "onpaar_van": "1", "onpaar_tot": "25", "paar_van": "2", "paar_tot": "60", "sector": "J200", "stadsdeel": "Gent-Zuid", "wijkNr": "27", "wijknaam": "Sint-Denijs-Westrem"}
```

2. Laat telkens de naam van het informatieveld en het daaropvolgende dubbelpunt weg. Verwijder ook de accolades aan het begin en einde van elke regel. Toegepast op het vorige voorbeeld wordt dit

```
1 | "9000", "2101", "Aaigemstraat", "", "", "", "A40", "Gent-Zuid", "22", "Stationsbuurt Noord"
2 | "9000", "7131", "Banierstraat", "", "", "2", "30", "E32", "Gent-Noordoost", "13", "Dampoort"
3 | "9051", "8511", "Broekkantstraat", "1", "25", "2", "60", "J200", "Gent-Zuid", "27", "Sint-Denijs-Westrem"
```

3. Verwijder de dubbele aanhalingstekens die rond de verschillende velden staan, en vervang de komma door een puntkomma als scheidingsteken. Toegepast op het vorige voorbeeld wordt dit

```
1 | 9000;2101;Aaigemstraat;;;A40;Gent-Zuid;22;Stationsbuurt Noord
2 | 9000;7131;Banierstraat;;;2;30;E32;Gent-Noordoost;13;Dampoort
3 | 9051;8511;Broekkantstraat;1;25;2;60;J200;Gent-Zuid;27;Sint-Denijs-Westrem
```

4. Verander de volgorde van de velden van `postcode`, `straatcode`, `straatnaam`, `onpaar_van`, `onpaar_tot`, `paar_van`, `paar_tot`, `sector`, `stadsdeel`, `wijkNr`, `wijknaam` naar `straatnaam`, `wijknaam`, `stadsdeel`, `postcode`, `onpaar_van`, `onpaar_tot`, `paar_van`, `paar_tot`. De velden `straatcode`, `sector` en `wijkNr` mogen weggelaten worden. Toegepast op het vorige voorbeeld wordt dit

```

1 | Aaigemstraat ; Stationsbuurt Noord ; Gent-Zuid ; 9000 ; ; ;
2 | Banierstraat ; Dampoort ; Gent-Noordoost ; 9000 ; ; ; 2 ; 30
3 | Broekkantstraat ; Sint-Denijs-Westrem ; Gent-Zuid ; 9051 ; 1 ; 25 ; 2 ; 60

```

5. Verwijder de regels waarvan het veld `onpaar_tot` (veld 6) ontbreekt, het veld `paar_tot` (veld 8) ontbreekt of waar beide velden ontbreken.
6. Sorteer de straten aflopend volgens `paar_tot`. Indien straten dezelfde `paar_tot` waarde hebben, sorteer dan aflopend volgens `onpaar_tot`. Indien ook `onpaar_tot` gelijk is, sorteer dan alfabetisch op `straatnaam`.

Opgave 3

(10 pt)

Het alfabet bestaat uit 26 karakters, terwijl het toetsenbord van een mobiele telefoon of een GPS doorgaans veel minder toetsen bevat. Om op dergelijke toestellen toch makkelijk woorden te kunnen intoetsen, werd oorspronkelijk gebruik gemaakt van de zogenaamde *MultiTap* techniek. Het principe is eenvoudig: op elke toets worden drie of meer letters afgebeeld (2=ABC, 3=DEF, 4=GHI, 5=JKL, 6=MNO, 7=PQRS, 8=TUV, 9=WXYZ). Om de gewenste letter te krijgen, moet je de toets waarop de letter wordt afgebeeld één of meer keer indrukken, afhankelijk van de positie van de letter op de toets.



Stel dat je bijvoorbeeld de letter A wilt typen. Dan moet je één keer op toets 2 drukken. Wil je echter een B typen, dan moet je twee keer kort na elkaar op toets 2 drukken. Om na elkaar twee letters te typen die op dezelfde toets worden afgebeeld, moet de gebruiker een korte pauze inlassen alvorens dezelfde toets opnieuw in te drukken. We zullen een spatie () gebruiken om een dergelijke pauze voor te stellen. Op die manier staat 2_2 voor AA, terwijl 22 staat voor B.

Elke regel van het tekstbestand `multitap.txt` bevat de multitapcodering van een woord dat enkel bestaat uit letters van het alfabet, gevolgd door een spatie en het woord zelf. Aangezien de multitapcodering zelf ook spaties kan bevatten, kan het dus gebeuren dat er meerdere spaties voorkomen op één enkele regel. Gevraagd wordt:

1. Bepaal reguliere expressies voor elk van onderstaande verzamelingen. Daarbij staat \mathcal{M} voor de verzameling van alle multitapcoderingen van woorden die enkel bestaat uit letters van het alfabet. Als m de multitapcodering van een woord voorstelt, dan stelt m' de multitapcodering voor waaruit de spaties verwijderd zijn. Probeer de reguliere expressies bovendien zo kort mogelijk te houden.

- (a) $\alpha = \{m \in \mathcal{M} \mid \text{bij elke spatie in } m \text{ staan er 2 posities daarvoor en 2 posities daarachter 2 verschillende cijfers waarvan de som gelijk is aan 8; als } m \text{ meerdere spaties heeft, dan zijn opeenvolgende spaties minstens gescheiden door 4 cijfers}\}$

voorbeelden: 222 266222 266 $\in \alpha$,

6444 44555 5777, 6444 44444 4477, 6444 444 4477 $\notin \alpha$

(b) $\beta = \{m \in \mathcal{M} \mid \text{opeenvolgende cijfers van } m' \text{ vormen een stijgende rij} \}$

voorbeelden: 22335566 66678 $\in \beta$,

444663 33997777 7 777666 664 $\notin \beta$

(c) $\gamma = \{m \in \mathcal{M} \mid m' \text{ bevat minstens 2 reeksen van minstens 4 opeenvolgende gelijke cijfers} \}$

voorbeelden: 55533 33777799992 2555, 777666877 77777 7833 3366 $\in \gamma$,

777444777744422266677733 388222844433 $\notin \gamma$

(d) $\delta = \{m \in \mathcal{M} \mid \text{elk cijfer van } m' \text{ is gelijk aan het vorige of volgende cijfer (of beide)} \}$

voorbeelden: 3366 666777 7777883 3 33777 $\in \delta$,

77778666 666 68882 27778 88444 4 $\notin \delta$

Gebruik een commando uit de **grep** familie om enkel die regels van het bestand **multitap.txt** te selecteren, waarvan de multitapcodering behoort 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 multitapcodering uit de verzameling $\alpha \cap \beta$

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

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

(d) het vierde woord staat op de unieke regel met multitapcodering 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.

Opgave 4

(6 pt)

Geef \LaTeX -code die precies hetzelfde resultaat oplevert als het tekstfragment in onderstaand kader. Zorg er daarbij voor dat de opmaak zo getrouw mogelijk behouden blijft. Plaats een PDF bestand met daarin het gecompileerde \LaTeX -fragment in het ZIP-bestand dat je indient via Indianio.

Het limietonderzoek van de som van twee reële functies wordt beschreven in onderstaande tabel.

$\lim_{x \rightarrow a} f(x)$	$\lim_{x \rightarrow a} g(x)$	$\lim_{x \rightarrow a} (f + g)(x)$
$\in \mathbb{R}$	$\in \mathbb{R}$	$\lim_{x \rightarrow a} f(x) + \lim_{x \rightarrow a} g(x)$
$\in \mathbb{R}$	$+\infty$	$+\infty$
$+\infty$	$+\infty$	$+\infty$
$-\infty$	$+\infty$	onbepaald
$\in \mathbb{R}$	$-\infty$	$-\infty$
$-\infty$	$-\infty$	$-\infty$

Opgave 5

(10 pt)

We hebben een afbeelding verknipt in smalle verticale stroken. Deze afbeeldingsstroken bevinden zich ergens op het Internet, elk met hun eigen URL. Je opdracht (*should you decide to accept it*) bestaat erin de URLs van de afbeeldingsstroken te achterhalen en er de originele afbeelding mee te reconstrueren.

De URLs van de afbeeldingsstroken werden verborgen in een apache logbestand (de open source apache webserver is de meest gebruikte server op het Internet). Elk logbestand komt van een bepaalde server, en de gezochte afbeeldingsstroken bevinden zich ergens op die server. De naam van de afbeelding en de hostname van de server waarop de afbeeldingsstroken zich bevinden, zitten gecodeerd in de naam van het logbestand: het logbestand `plaats_www.voorbeeld.org` komt bijvoorbeeld van de webserver `www.voorbeeld.org` (formeel zeggen we dat de servernaam volgt op de eerste underscore in de naam van het logbestand). Dit logbestand bevat voorts de informatie die nodig is om de afbeelding `plaats` (het gedeelte voor de underscore in de bestandsnaam) te reconstrueren. Onderstaand voorbeeld geeft aan hoe de informatie in een apache logbestand er uitziet:

```
10.254.254.8 - - [06/Aug/2007:00:14:08 -0700] "GET /foo/bar/puzzel/bar-abab-baaa.jpg
HTTP/1.1" 200 5910 "-" "Mozilla/5.0 (X11; U; Linux i686 (x86_64); en-US; rv
:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4"
```

De eerste reeks getallen stelt het IP-adres voor van de computer die een vraag gericht heeft aan de webserver. Het deel "GET *pad* HTTP" is voor ons het meest interessant, omdat daarin het pad voorkomt van de URL waarvoor een vraag gericht werd aan de webserver. Dit pad bevat zelf nooit spaties, en het wordt gescheiden van GET en HTTP door spaties.

In het logbestand hebben de afbeeldingsstroken een padnaam die de term `puzzel` bevat. Combineer de padnaam van elke URL met de hostname van de server om de volledige URL van de afbeeldingsstrook te bekomen, bv. `http://www.voorbeeld.org/foo/bar/puzzel/bar-abab-baaa.jpg`. URLs die meerdere keren voorkomen, moeten ontduddeld worden. De bestandsnaam van elke afbeeldingsstrook bevat ook één of meer koppelteken (-). De URLs moeten alfabetisch gerangschikt worden op basis van het gedeelte na het laatste koppelteken. Hierbij mag geen onderscheid gemaakt worden tussen hoofdletters en kleine letters. De sorteersleutel van bovenstaande URL is dan bijvoorbeeld `baaa.jpg`.

Schrijf een `bash` shell script `logstroken` waaraan de locatie van een logbestand moet doorgegeven worden. In dat logbestand zitten de stroken van een afbeelding verborgen. Voor bijvoorbeeld het logbestand `plaats_users.ugent.be` moet het shell script een geldig XHTML bestand `plaats.html` (deze naam is dus opgebouwd op basis van het gedeelte voor de underscore in de bestandsnaam van het logbestand) aanmaken in de huidige directory. Dit XHTML bestand moet de originele afbeelding — samengesteld uit de verschillende afbeeldingsstroken — horizontaal gecentreerd in het browservenster weergeven. De volledige afbeelding moet omgeven worden door een dikke rode rand van 5 pixels breed. Gebruik CSS-opmaak op documentniveau om de correcte weergave van de afbeelding vast te leggen. Ter controle kan je je script ook nog uittesten met het logbestand `dier_computergebruik.ugent.be` dat met het examen werd meegeleverd.

