

---

---

## EXAMEN: Computergebruik

---

---

1<sup>e</sup> Bachelor Informatica  
prof. dr. Peter Dawyndt  
groep 2

woensdag 09-01-2013, 14:00  
academiejaar 2012-2013  
eerste zittijd

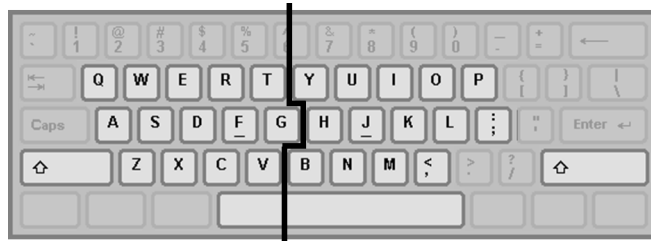
### Opgave 1

(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. Op onderstaand toetsenbord met *regular US* indeling geeft de dikke zwarte lijn aan welke toetsen doorgaans met de linker- en rechterhand worden aangeslagen. Enkel de duidelijk weergegeven toetsen zijn relevant voor deze opgave, met 10 toetsen op de eerste en de tweede rij, en 8 toetsen op de derde rij. De shift toets (⇧) wordt gebruikt om hoofdletters (met shift) en kleine letter (zonder shift) weer te geven, of de karakters bovenaan — resp. onderaan — de toets.

Iemand heeft op dit toetsenbord een regel tekst getypt, maar heeft daarbij zijn twee handen omgewisseld (je moet het maar eens proberen). Als hij alle karakters op de eerste rij achtereenvolgens zou intypen met de shift toets ingedrukt, dan bekommt hij dus `YUIOPQWERT` in plaats van `QWERTYUIOP`. Als hij analoog alle karakters op de derde rij achter elkaar intypt zonder de shift toets in te drukken, dan bekommt hij `bnm,zxv` in plaats van `zxcvbnm`, (inclusief de komma). Het omwisselen van de handen heeft geen invloed op het intypen van spaties.



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 "Aiffr urofK" | ...  
Hello world  
$ echo "Dxwpa ej cq aior" | ...  
Knuth is my hero
```

2. Onder de gegeven directory `testdir` zitten een reeks gewone bestanden en subdirectories op verschillende niveau's diep. Alle bestandsnamen van directories en gewone bestanden bestaan enkel uit klein letters, en optioneel één enkele underscore (`_`). Gevraagd wordt:
  - (a) Geef een commando dat underscores verwijdert uit de namen van alle gewone bestanden onder de directory `testdir` (dus inclusief alle onderliggende subdirectories).
  - (b) Geef een commando dat underscores verwijdert uit de namen van alle gewone bestanden én subdirectories onder de directory `testdir` (dus inclusief alle onderliggende subdirectories).

3. Een natuurlijk getal  $k$  dat bestaat uit  $n$  cijfers en dat gelijk is aan de som van de  $n$ -de machten van zijn cijfers wordt een  *$n$ -narcistisch getal* genoemd. Enkele voorbeelden:

$$\begin{aligned}153 &= 1^3 + 5^3 + 3^3 \\1634 &= 1^4 + 6^4 + 3^4 + 4^4 \\54748 &= 5^5 + 4^5 + 7^5 + 4^5 + 8^5 \\548834 &= 5^6 + 4^6 + 8^6 + 8^6 + 3^6 + 4^6 \\1741725 &= 1^7 + 7^7 + 4^7 + 1^7 + 7^7 + 2^7 + 5^7\end{aligned}$$

Schrijf een **bash** shell script **narcistisch** waaraan twee argumenten moeten doorgegeven worden:  $k$  en  $n$ . Het shell script moet de som van de  $n$ -de machten van de cijfers van  $k$  naar standaard uitvoer uitschrijven. De ideale oplossing is een shell script dat bestaat uit één enkel commando en geen tijdelijke bestanden aanmaakt. 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 het script deze tijdelijke bestanden finaal terug verwijdert).

```
$ narcistisch 153 3
153
$ narcistisch 1634 4
1634
$ narcistisch 54748 5
54748
$ narcistisch 548834 6
548834
$ narcistisch 1741725 7
1741725
$ narcistisch 123456789 3
2025
$ narcistisch 123456789 9
574304985
```

4. Een *woordkubus* is een spelletje waarbij woorden moeten gevormd worden met de negen letters die in een  $3 \times 3$  rooster staan. Elke letter van het rooster mag slechts één keer gebruikt worden. Er bestaat altijd een woord dat alle letters van het rooster gebruikt. Woorden moeten minstens uit vier letters bestaan en moeten de centrale letter van het rooster bevatten. Wie het meeste woorden kan vormen, wint. Er kunnen bijvoorbeeld drie woorden gevormd worden voor onderstaande opgave: **urinezuur**, **Uzun** en **zuur**.

E	I	U
N	U	R
R	U	Z

Schrijf een **bash** shell script **woordkubus** waaraan negen kleine letters als argument moeten doorgegeven worden. Deze letters vormen de opgave van een woordkubus, uitgelezen uit het rooster van links naar rechts, en van boven naar onder. Het script moet alle mogelijke oplossingen van de gegeven woordkubus uitschrijven naar standaard uitvoer, gesorteerd in alfabetische volgorde. Deze oplossingen moeten gezocht worden in het bestand **words** (op **helios** te vinden op **/usr/share/dict/words**) dat een lijst van woorden bevat, elk op een afzonderlijke regel. Zowel bij het sorteren als bij het zoeken van de oplossingen mag geen onderscheid gemaakt worden tussen hoofdletters en kleine letters. De woorden moeten wel hun oorspronkelijke schrijfwijze uit het bestand behouden.

```

$ woordkubus e i u n u r r u z
urinezuur
Uzun
zuur
$ woordkubus u a d r h r a u v
haar
haard
haardvuur
hard
huur
Radha
vuurhaard
$ woordkubus d e e i i j r w z
Jiri
weiderij
wiedijzer
ziederij

```

De ideale oplossing is een shell script dat bestaat uit één enkel commando en geen tijdelijke bestanden aanmaakt. 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 het script deze tijdelijke bestanden finaal terug verwijdert).

**Tip:** Een mogelijke oplossing bestaat erin het woord op elke regel van het bestand **words** te verdubbelen (bv. **urinezuur** wordt **urinezuur urinezuur**) en daarna de letters uit de woordkubus te schrappen uit het eerste woord. Regels waarvan alle letters van het eerste woord geschrapt zijn, vormen dan een kandidaat-oplossing van de woordkubus. Merk op dat oplossingen ook minstens vier letters moeten hebben en de centrale letter van de woordkubus moeten bevatten.

## Opgave 2

(12 pt)

Gegeven is een tekstbestand **elements.txt** waarin je de informatie over alle chemische elementen in JSON-formaat kan vinden. Dit bestand begint met een open accolade {, gevolgd door een lijst van atoomobjecten gescheiden door een komma, gevolgd door een sluitende accolade }. Elk atoomobject begin met de naam van het element, gevolgd door : {, gevolgd door een lijst van eigenschappen gescheiden door een komma. Deze eigenschappen hebben telkens het formaat "**naam veld**" : "**waarde veld**". 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 **elements.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 element slechts 1 regel gebruikt. Verwijder hiervoor de nodige newlines, vervang opeenvolgende spaties door een enkele spatie en verwijder eventuele spaties aan het begin van een regel. Verwijder ook de eerste en laatste regel van het bestand (de regels die de accolades bevatten). Bijvoorbeeld:

```

1 | {
2 |   "Hydrogen" : {
3 |     "symbol" : "H",
4 |     "atomic_number" : 1,
5 |     "atomic_weight" : 1.00794,
6 |     "melting_point K" : 14.01,
7 |     "boiling_point K" : 20.28,
8 |     "atomic_radius pm" : 79,
9 |     "covalent_radius pm" : 32,
10 |    "ionic_radius pm" : "",

```

```

11     "atomic_volume cm3/mol" : 14.1,
12     "fusion_heat (kJ/mol)" : "0.117 (H-H)",
13     "evaporation_heat (kJ/mol)" : "0.904 (H-H)",
14     "pauling_negativity" : 2.20,
15     "first_ionizing kJ/mol" : 1311.3,
16     "lattice_structure" : "HEX",
17     "lattice_constant ang" : 3.750
18 },
19 "Helium" : {
20     "symbol" : "He",
21     "atomic_number" : 2,
22     "atomic_weight" : 4.002602,
23     "melting_point K" : 0.95,
24     "boiling_point K" : 4.216,
25     "atomic_radius pm" : 0.0,
26     "covalent_radius pm" : "-",
27     "ionic_radius pm" : "",
28     "atomic_volume cm3/mol" : 31.8,
29     "fusion_heat (kJ/mol)" : "-",
30     "evaporation_heat (kJ/mol)" : 0.08,
31     "pauling_negativity" : "-",
32     "first_ionizing kJ/mol" : 2361.3,
33     "lattice_structure" : "HEX",
34     "lattice_constant ang" : 3.570
35 }
36 }

```

wordt omgezet naar

```

1 "Hydrogen" : { "symbol" : "H", "atomic_number" : 1, "atomic_weight" : 1.00794, "
melting_point K" : 14.01, "boiling_point K" : 20.28, "atomic_radius pm" : 79, "
covalent_radius pm" : 32, "ionic_radius pm" : "", "atomic_volume cm3/mol" : 14.1, "
fusion_heat (kJ/mol)" : "0.117 (H-H)", "evaporation_heat (kJ/mol)" : "0.904 (H-H)", "
pauling_negativity" : 2.20, "first_ionizing kJ/mol" : 1311.3, "lattice_structure" : "HEX
", "lattice_constant ang" : 3.750 },
2 "Helium" : { "symbol" : "He", "atomic_number" : 2, "atomic_weight" : 4.002602, "
melting_point K" : 0.95, "boiling_point K" : 4.216, "atomic_radius pm" : 0.0, "
covalent_radius pm" : "-", "ionic_radius pm" : "", "atomic_volume cm3/mol" : 31.8, "
fusion_heat (kJ/mol)" : "-", "evaporation_heat (kJ/mol)" : 0.08, "pauling_negativity" :
"-", "first_ionizing kJ/mol" : 2361.3, "lattice_structure" : "HEX", "lattice_constant ang
" : 3.570 }

```

2. Laat telkens de naam van de eigenschap en het daaropvolgend dubbelpunt weg. Verwijder ook de accolade aan het begin van elke lijst eigenschappen en de accolade en eventuele komma op het einde van de regel. Toegepast op het vorige voorbeeld wordt dit

```

1 "Hydrogen" : "H", 1, 1.00794, 14.01, 20.28, 79, 32, "", 14.1, "0.117 (H-H)", "0.904 (H-H)
", 2.20, 1311.3, "HEX", 3.750
2 "Helium" : "He", 2, 4.002602, 0.95, 4.216, 0.0, "-", "", 31.8, "-", 0.08, "-", 2361.3, "
HEX", 3.570

```

3. Vervang het dubbelpunt na de naam van het element, en de komma's tussen de informatievelden door een puntkomma zodat dit het scheidingsteken wordt. Verwijder ook eventuele spaties rond de elementnaam en informatievelden en verwijder overal de dubbele aanhalingstekens. Toegepast op het vorige voorbeeld wordt dit

```

1 Hydrogen;H;1;1.00794;14.01;20.28;79;32;;14.1;0.117 (H-H);0.904 (H-H);2.20;1311.3;HEX
;3.750
2 Helium;He;2;4.002602;0.95;4.216;0.0;-;;31.8;-;0.08;-;2361.3;HEX;3.570

```

4. Verwijder de regels waarvan het laatste veld (`lattice_constant ang`) niet numeriek is (bv. n/a of -).
5. Sorteer aflopend op het laatste veld (`lattice_constant ang`). Sorteer bij gelijke waarde alfabetisch op het eerste veld.

## 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)  $\alpha = \{m \in \mathcal{M} \mid \text{oneven cijfers komen steeds in paren voor in } m'\}$

voorbeelden: 66622244899663\_33\_3444844455  $\in \alpha$ ,  
44666 6664 44278444 433777  $\notin \alpha$

- (b)  $\beta = \{m \in \mathcal{M} \mid \text{opvolgende cijfers van } m' \text{ zijn ofwel gelijk, ofwel is het ene even en het andere oneven}\}$

voorbeelden: 47776663344433663 33  $\in \beta$ ,  
77774443377792 27773 33  $\notin \beta$

- (c)  $\gamma = \{m \in \mathcal{M} \mid m' \text{ bevat een geïsoleerd cijfer } 3\}$

**uitleg:** een cijfer is geïsoleerd als de cijfers die eventueel voor of na het cijfer staan verschillen van dat cijfer

voorbeelden: 37772 2224483366, 8777335537776667, 87776668777744334443  $\in \gamma$ ,  
3338664337773366  $\notin \gamma$

- (d)  $\delta = \{m \in \mathcal{M} \mid m' \text{ bevat maximaal twee paren opvolgende gelijke cijfers}\}$

opmerking: bij de definitie van  $\delta$  moeten overlappende paren in rekening gebracht worden: de reeks 222 telt dus twee paren opvolgende gelijke cijfers, en de reeks 7777 telt drie paren opvolgende gelijke cijfers.

voorbeelden: 62 287255  $\in \delta$ ,  
552 22 25558, 43396663, 2999924  $\notin \delta$

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  $\alpha$ ,  $\beta$ ,  $\gamma$  en  $\delta$  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  $\text{\LaTeX}$ -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  $\text{\LaTeX}$ -fragment in het ZIP-bestand dat je indient via Indianio.

Het limietonderzoek van het product van twee reële functies wordt beschreven in onderstaande tabel.

$\lim_{x \rightarrow a} f(x)$	$\lim_{x \rightarrow a} g(x)$	$\lim_{x \rightarrow a} fg(x)$
$\in \mathbb{R}$	$\in \mathbb{R}$	$\lim_{x \rightarrow a} f(x) \cdot \lim_{x \rightarrow a} g(x)$
$\in ]0, +\infty]$	$+\infty$	$+\infty$
$\in [-\infty, 0[$	$+\infty$	$-\infty$
0	$+\infty$	onbepaald
$\in ]0, +\infty]$	$-\infty$	$-\infty$
$\in [-\infty, 0[$	$-\infty$	$+\infty$
0	$-\infty$	onbepaald

## 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.

