
Computergebruik: EXAMEN

1^e Bachelor Informatica
prof. dr. Peter Dawyndt
groep 2

woensdag 30-08-2017, 14:00
academiejaar 2016-2017
tweede zittijd

Opgave 1 (populaire voornamen)

(3 pt)

Een bekend hackerscollectief heeft de server gekraakt waarop de kieslijst voor de rectorverkiezingen van de Universiteit Gent bewaard wordt. Daarop vonden ze een directory met een diepgeneste structuur van subdirectories. Verspreid over al die subdirectories staan een rits kleine tekstbestanden met daarin de gegevens van studenten en medewerkers die op de kieslijst voorkomen. Je kan de inhoud van deze directory terugvinden in het ZIP-bestand `kieslijst.zip`.

De naam van elk tekstbestand wordt gevormd door het eerste karakter van de familienaam van alle studenten en medewerkers uit het bestand, gevolgd door een underscore (`_`), een code van twee hoofdletters die de faculteit aanduidt van de studenten en medewerkers uit het bestand (bijvoorbeeld `WE` voor de faculteit Wetenschappen of `GE` voor de faculteit Geneeskunde en Gezondheidswetenschappen) en de extensie `.txt`. Hieronder zie je bijvoorbeeld de inhoud van het tekstbestand `Q_GE.txt` met daarin alle studenten en medewerkers op de kieslijst van de faculteit Geneeskunde en Gezondheidswetenschappen wiens familienaam begint met de letter `Q`.

```
000161167116;Quintens;Vincent
000160983927;Qian;Vincent
000130866437;Quintyn;Marvin
000130722351;Quatacker;Elise
000110162997;Qu;Sophie
000110703167;Quintens;Dries
000140984446;Quaghebeur;Stephanie
000151046477;Quinteiro González;Kaat
000150198133;Quaghebeur;Liza
```

Elke regel van zo een tekstbestand bevat informatie van één enkele student of medewerker van de UGent die op de kieslijst voorkomt. Deze informatie bestaat uit de volgende drie informatievelen, die telkens van elkaar gescheiden worden door een puntkomma (`;`): *i*) UGent ID, *ii*) familienaam en *iii*) voornaam.

Opgave

We hebben het ZIP-bestand `kieslijst.zip` uitgepakt in de huidige directory (die initieel leeg was). Geef een Unix commando dat op basis van de inhoud van het uitgepakte ZIP-bestand een overzicht uitschrijft op `stdout` met daarin de top 5 van de meest voorkomende voornamen van alle studenten en medewerkers op de kieslijst van de faculteit Wetenschappen, wiens familienaam begint met de letter `Q` en die voor het eerst zijn ingeschreven in het academiejaar 2016-2017 (UGent ID begint met `00016`). Het overzicht moet bestaan uit twee kolommen met respectievelijk het aantal voorkomens en een voornaam, van elkaar gescheiden door één enkele spatie. Het overzicht moet gerangschikt worden, eerst op dalend aantal voorkomens, en dan alfabetisch op voornaam. Het uiteindelijke resultaat moet er als volgt uitzien:

```
7 Robin
6 Robbe
6 Simon
6 Tibo
6 Victor
```

Opgave 2 (formule van Leibniz voor π)

(3 pt)

Gottfried Leibniz vond de volgende benadering voor het getal π :

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Let op het feit dat de formule alterneert tussen de optelling en de aftrekking van termen.

Opgave

Vul volgende Unix commandolijn aan, zodat een benadering van het getal π naar **stdout** wordt uitgeschreven, uitgaande van de formule van Leibniz waarbij de laatste term gelijk is aan $\frac{4}{i}$ (als i oneven is) of aan $\frac{4}{i-1}$ (als i even is).

```
$ echo i | ...
```

In het specifieke geval van $i = 19$ moet het commando

```
$ echo 19 | ...
3.04183961892940221113
```

bijvoorbeeld het resultaat uitschrijven van de bewerking

$$\frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} - \frac{4}{15} + \frac{4}{17} - \frac{4}{19}$$

Opgave 3 (zeg het met Adele)

(4 pt)

De beste manier om online fraudeurs te slim af te zijn, is om hen meteen te blokkeren. Maar er zijn ook grappige manieren om duidelijk te maken dat je niet gediend bent met iemands vraag naar geld. Dat bewees Frank Fleming Jensen. De Deen kreeg via Facebook een contactverzoek van iemand "die stapelverliefd op hem was" en enkel nog geld nodig had voor een vliegtuigticket om bij hem te geraken. Fleming Jensen vond er niets beters op dan te antwoorden met stukjes tekst uit "Hello", de wereldhit van Adele. En dat had de persoon aan de andere kant van de conversatie niet onmiddellijk door. Tot zij (of hij) het wel snapte en het huis te klein was ("I smash your life"). Fleming Jensen zelf vond het hilarisch en publiceerde de dialoog op zijn Facebookpagina. Fleming Jensen op de Deense nieuwswebsite MetroXpress:

Het gebeurde allemaal heel spontaan. Ik had zonet het nummer "Hello" op de radio gehoord, toen iemand me online contacteerde. Het was mega-bevredigend. Ik kon me niet inhouden en brulde van het lachen achter mijn computer. Tegelijkertijd verbaasde het me wel dat de persoon aan de andere kant maar wanhopig bleef proberen om me geld af te troggelen. Het gaf me zo'n goed gevoel dat ik die persoon kon terug-bedriegen.

Opgave

Gevraagd wordt om een bash shell script **mix** te schrijven waaraan twee locaties van tekstbestanden moeten doorgegeven worden die evenveel regels tekst bevatten (het shell script mag ervan uitgaan dat de doorgegeven argumenten geldig zijn, en moet dit dus niet expliciet controleren). Het shell script moet de regels van de twee bestanden geschrinkt uitschrijven naar **stdout**: eerst de eerste regel uit het bestand dat als eerste argument werd doorgegeven, dan de eerste regel uit het bestand dat als tweede argument werd doorgegeven, dan de tweede regel uit het bestand dat als eerste argument werd doorgegeven, dan de tweede regel uit het bestand dat als tweede argument werd doorgegeven, enzoverder. Alle regels uit het tweede bestand moeten ingesloten worden tussen >>> en <<<. Als we bijvoorbeeld dit tekstfragment van 4 regels uit het nummer "Martha" van Tom Waits:

```
Operator, number, please
It's been so many years
Will she remember my old voice
While I fight the tears?
```

mixen met dit tekstfragment van 4 regels uit het nummer "Hello" van Adele:

```
Hello from the other side
I must have called a thousand times
To tell you I'm sorry for everything that I've done
But when I call you never seem to be home
```

dan krijgen we de volgende 8 regels als resultaat:

```
Operator, number, please
>>>Hello from the other side<<<
It's been so many years
>>>I must have called a thousand times<<<
Will she remember my old voice
>>>To tell you I'm sorry for everything that I've done<<<
While I fight the tears?
>>>But when I call you never seem to be home<<<
```

Idealiter bestaat het shell script uit één enkel Unix commando.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat de tekstbestanden `tom_waits.txt` en `adele.txt` zich in de huidige directory bevinden.

```
$ mix tom_waits.txt adele.txt
Operator, number, please
>>>Hello from the other side<<<
It's been so many years
>>>I must have called a thousand times<<<
Will she remember my old voice
>>>To tell you I'm sorry for everything that I've done<<<
While I fight the tears?
>>>But when I call you never seem to be home<<<
```

Alle bestanden die gebruikt worden bij het testen van je ingediende oplossing, kan je terugvinden in het ZIP-bestand `adele.zip`.

Opgave 4 (Baconversleuteling)

(10 pt)

In 1605 ontwikkelde de Britse filosoof, wetenschapper en politicus Francis Bacon een versleutelingsmethode die in twee stappen werkt.

A Baconian cipher key showing the mapping of letters A through Z to binary strings of 'a' and 'b'. The letters are arranged in four rows of six. The strings are as follows:

A	B	C	D	E	F
aaaba	aaaab	aaaba	aaabb	aabaa	aabab
G	H	I	K	L	M
aabba	aabbb	abaaa	abaab	ababa	ababb
N	O	P	Q	R	S
abbaa	abbab	abbba	abbbb	baaaa	baaab
T	V	W	X	Y	Z
baaba	baabb	babaa	babab	babba	babbb

Bij het coderen wordt elke letter van het originele bericht eerst vertaald naar een groep van vijf letters uit een tweeletter-alfabet (bijvoorbeeld a en b). In bovenstaande figuur zie je bijvoorbeeld de sleutel met het tweeletter-alfabet (a en b) die Francis Bacon gebruikte in zijn werk *De Augmentis Scientiarum* (1605). Omdat het ging om een werk in het Latijn, was het toenertijd gebruikelijk om de letters i en j aan elkaar gelijk te stellen, net zoals de letters u en v. Aangezien het gaat om een binaire codering, die gebruik maakt van twee symbolen op vijf posities, kunnen er dus in totaal $2^5 = 32$ symbolen gecodeerd worden. Het is dus perfect mogelijk om de Baconversleuteling toe te passen met een sleutel die elk van de 26 letters uit ons alfabet op een unieke manier codeert, en bovendien nog ruimte laat voor 6 extra karakters (bijvoorbeeld een spatie en enkele leestekens).



In tweede instantie wordt een willekeurige tekst genomen of verzonnen, die wordt uitgeschreven aan de hand van twee verschillende lettertypes. Het ene lettertype komt overeen met de letter a, en het andere lettertype met de letter b uit de vorige stap. Hierbij maakt het dus niet uit welke letter gebruikt wordt, enkel het lettertype wordt gebruikt voor de code. In bovenstaande figuur zie je de lettertypes die Francis Bacon gebruikte. Om te verhullen dat het om een gecodeerd bericht ging, maakte hij gebruik van twee lettertypes die vaak slechts op een subtiele manier van elkaar verschillen.

In plaats van te spelen met lettertypes, zou je bijvoorbeeld ook de letter a kunnen voorstellen door kleine letters, en de letter b door hoofdletters. Laat ons dat eens toepassen in een voorbeeld waarin we het woord ALICE willen versleutelen. In eerste instantie vervangen we elke letter door een reeks van vijf a's of b's:

A L I C E
 aaaaa ababa abaaa aaaba aabaa

Daarna coderen we deze boodschap in de tekst *Draco Dormiens Numquam Titillandus*:

aaaaa ababa abaaa aaaba aabaa
 draco dOrMI eNsnu mquAm tiTil

Om het principe nog duidelijker te illustreren hebben we hierbij alle hoofdletters en de corresponderende letters b in het **vet** weergegeven. In een handschrift valt een codering met twee nagenoeg gelijke lettertypes haast niet op.

Opgave

Het tekstbestand `bacon.txt` bevat de baconversleuteling van een reeks woorden (die enkel bestaan uit kleine letters). Elke regel van het bestand bevat de baconversleuteling van een woord, een spatie en het originele woord dat versleuteld werd. De baconversleuteling maakt gebruik van hoofdletters en kleine letters om respectievelijk de letters `b` en `a` uit de versleuteling voor te stellen.

In de rest van deze opgave gebruiken we de term letter als we expliciet geen onderscheid willen maken tussen hoofdletters en kleine letters. Gevraagd wordt:

1. Bepaal reguliere expressies voor elk van onderstaande verzamelingen. Daarbij staat \mathcal{B} voor de verzameling van alle mogelijke baconversleutelingen van woorden, of met andere woorden voor alle reeksen van letters waarvan de lengte een veelvoud is van vijf.

- (a) $\alpha = \{b \in \mathcal{B} \mid \text{als je alle letters uit } b \text{ weglaat die geen A, B, C, O of N zijn, dan spel je de letters van het woord BACON}\}$

voorbeelden: `phuBUwYhdzHxgJmSatMPsUEIvWmzjGIqlCEKonHl distorts` $\in \alpha$,
`uzsIghksxnKeqNCJcaDDgBtekmMBxXxeDIk catting` $\notin \alpha$

- (b) $\beta = \{b \in \mathcal{B} \mid b \text{ bevat een reeks van minstens vier opeenvolgende hoofdletters die ingesloten zit tussen dezelfde kleine letter}\}$

voorbeelden: `GnlCyycqHeTilaZoLyjmvLGqpiWHOQTTinm scrimpy` $\in \beta$,
`oLRhuiaiaabaAllryJZzXGyhQnfymfvEymgeKAgF maintain` $\notin \beta$

- (c) $\gamma = \{b \in \mathcal{B} \mid b \text{ bestaat uit alternerende groepen van twee klinkers en twee medeklinkers}\}$

voorbeelden: `FniIfReEpviAZqiuSQiaxxIolYuamFeOhDIoWQEugYAUtxiOQs summerlong` $\in \gamma$,
`ZveSoxrCuqhjwSuCcsUCHLpglyHlQNmPgtgHuuVSfBiitmlTitRsmMh sectilities` $\notin \gamma$

- (d) $\delta = \{b \in \mathcal{B} \mid \text{elke letter komt hoogstens drie keer voor in } b\}$

voorbeelden: `HsaNWhbCMJvYygvnrnyZcdMrKaEwjRbhtFguJfqD0 thickset` $\in \delta$,
`ufbvZgImpkQsrUxuFDIrvDXWFdxEGQhYtddaWcTN basophil` $\notin \delta$

Gebruik een commando uit de `grep` familie om enkel die regels van het bestand `bacon.txt` te selecteren 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.

Opmerking: Gebruik voor deze opgave een recente versie van GNU `grep`. Op `helios` is een recente versie van GNU `grep` geïnstalleerd, maar Mac OS X gebruikt standaard typisch een oudere versie van GNU `grep`. Mac gebruikers kunnen voor de zekerheid dus best hun `grep` versie updaten naar de meest recente versie.

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 uit de verzameling $\alpha \cap \beta$

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

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

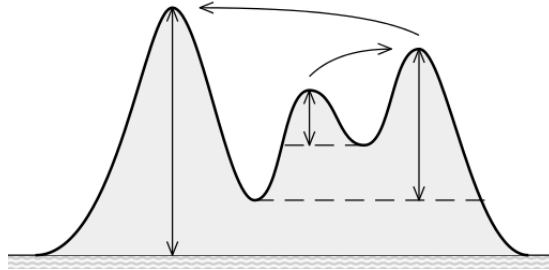
- (d) het vierde woord staat op de unieke regel uit de verzameling $\delta \cap \alpha$

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 5 (hoogste bergen op Aarde)

(10 pt)

De Mount Everest (ook bekend als *Qomolungma*, *Qomolangma* of *Sagarmatha Chomolungma*) is een berg in de Himalaya, op de grens van Nepal en China. De Mount Everest steekt 8848 meter (29029 voet) boven zeeniveau uit. Deze **absolute hoogte** is de meest geaccepteerde manier om hoogtes van bergen te meten en in die zin is de Mount Everest de hoogste berg op aarde.



De (topografische) prominentie of relatieve hoogte is een alternatieve definitie voor de onafhankelijke hoogte van een bergtop. De prominentie van een berg is het hoogteverschil tussen de top en de laagste hoogtelijn die de top helemaal omgeeft, maar die geen hogere top omgeeft, zoals je kan zien op bovenstaande figuur. Wanneer de prominentie van een top erg klein is, wordt deze top vaak niet gerekend als een echte berg. Dit geldt vaak voor zijtoppen van hoge bergen, bijvoorbeeld de zuidtop van de Mount Everest die geen echte berg is, ondanks de grote hoogte (8749 m, prominentie 10 m).

Opgave

Gegeven is een tekstbestand `bergen.txt` dat een lijst met de hoogste bergen op Aarde bevat. Elke regel van het bestand bevat informatie over één berg. Deze informatie bestaat uit de volgende elf velden, die van elkaar worden gescheiden door verticale strepen (|): *i*) volgnummer, *ii*) naam of namen, *iii*) absolute hoogte (in meter), *iv*) absolute hoogte (in voet), *v*) relatieve hoogte (in meter), *vi*) bergketen, *vii*) coördinaten, *viii*) ouder, *ix*) jaartal van eerste beklimming, *x*) aantal beklimmingen en *xi*) land of landen. De eerste regel van een bestand is een hoofding, waarvan de kopteksten ook telkens van elkaar gescheiden worden door verticale strepen.

Gevraagd wordt om — gebruik makend van de teksteditors `vi` of `vim` — een reeks 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 opdrachten moeten na elkaar uitgevoerd worden. Geen enkele van de commando's mag de hoofding van het bestand wijzigen, tenzij expliciet anders vermeld. Ter controle kan je gebruik maken van de meegeleverde bestanden `bergen.i.txt` (zip), die telkens de inhoud van het bestand bevatten nadat de *i*-de opdracht werd uitgevoerd.

1. Indien het veld met het jaartal van de eerste beklimming leeg is, dan moet het ingevuld worden met de waarde `unclimbed`. Zo moet

```
1 | Rank|Mountain|Height (m)|Height (ft)|Prominence (m)|Range|Coordinates|Parent mountain|First ascent|Ascents (failed
  | attempts)|Country
2 | 1|Mount Everest/Sagarmatha/Chomolungma|8,848|29,029|8,848|Mahalangur Himalaya|27°59'17"N 86°55'31"E|none|1953|>>145
  | (121)|Nepal/China
3 | 2|K2/Qogir/Godwin Austen|8,611|28,251|4,020|Baltoro Karakoram|35°52'53"N 76°30'48"E|Mount Everest|1954|45 (44)|Pakistan
  |/China
4 | ...
5 | 40|Gangkhar Puensum|7,570|24,836|2,995|Kula Kangri Himalaya|28°02'50"N 90°27'19"E|Kangchenjunga||0 (3)|Bhutan/China
6 | ...
7 | 93|Labuche Kang III|East|7,250|23,786|570|Labuche Himalaya|28°18'05"N 86°23'02"E|Labuche Kang||0 (0)|China
8 | ...
```

bijvoorbeeld omgezet worden naar (`bergen.1.txt`)

```
1 | Rank|Mountain|Height (m)|Height (ft)|Prominence (m)|Range|Coordinates|Parent mountain|First ascent|Ascents (failed
  | attempts)|Country
2 | 1|Mount Everest/Sagarmatha/Chomolungma|8,848|29,029|8,848|Mahalangur Himalaya|27°59'17"N 86°55'31"E|none|1953|>>145
  | (121)|Nepal/China
3 | 2|K2/Qogir/Godwin Austen|8,611|28,251|4,020|Baltoro Karakoram|35°52'53"N 76°30'48"E|Mount Everest|1954|45 (44)|Pakistan
  |/China
```

```

4 ...
5 40|Gangkhar Puensum|7,570|24,836|2,995|Kula Kangri Himalaya|28°02'50"N 90°27'19"E|Kangchenjunga|unclimbed|0 (3)|Bhutan/
  China
6 ...
7 93|Labuche Kang III/East|7,250|23,786|570|Labuche Himalaya|28°18'05"N 86°23'02"E|Labuche Kang|unclimbed|0 (0)|China
8 ...

```

- Het veld met het aantal beklimmingen eindigt optioneel op een spatie en een tekstfragment tussen ronde haken dat aangeeft hoeveel pogingen om de berg te beklimmen gefaald zijn. Je mag er van uitgaan dat er hoogstens één paar ronde haken voorkomt in het veld, en dat de haken altijd gebalanceerd zijn. Verplaats het aantal gefaalde beklimmingen (het optionele tekstfragment tussen ronde haken) naar een nieuw veld dat volgt op het veld met het aantal beklimmingen (spatie en tekst tussen ronde haken moeten uit het oorspronkelijke veld verwijderd worden). Voor bergen waarvoor het aantal beklimmingen niet vermeld wordt, moet ook een nieuw (leeg) veld op die positie aangemaakt worden. Ook de hoofding van het veld met het aantal beklimmingen moet over twee velden gesplitst worden, zoals aangegeven in onderstaand voorbeeld. We krijgen dan (`bergen.2.txt`)

```

1 Rank|Mountain|Height (m)|Height (ft)|Prominence (m)|Range|Coordinates|Parent mountain|First ascent|Ascents|failed
  attempts|Country
2 1|Mount Everest/Sagarmatha/Chomolungma|8,848|29,029|8,848|Mahalangur Himalaya|27°59'17"N 86°55'31"E|none
  |1953|>>145|121|Nepal/China
3 2|K2/Qogir/Godwin Austen|8,611|28,251|4,020|Baltoro Karakoram|35°52'53"N 76°30'48"E|Mount Everest|1954|45|44|Pakistan/
  China
4 ...
5 49|Muztagh Ata|7,509|24,636|2,698|Muztagata (Eastern Pamirs)|38°16'33"N 75°06'58"E|Kongur Tagh|1956|Many||China
6 50|Ismoil Somoni Peak|7,495|24,590|3,402|Pamir (Academy of Sciences Range)|38°56'35"N 72°00'57"E|Muztagh Ata|1933|||
  Tajikistan
7 ...

```

- De drie velden die de hoogte van de berg aangeven, bevatten natuurlijke getallen waarvan de groepen duizendtallen door komma's van elkaar gescheiden worden. Zorg ervoor dat die komma's verwijderd worden. Je mag er hierbij van uitgaan dat in het bestand enkel komma's voorkomen in deze drie velden. We krijgen dan (`bergen.3.txt`)

```

1 Rank|Mountain|Height (m)|Height (ft)|Prominence (m)|Range|Coordinates|Parent mountain|First ascent|Ascents|failed
  attempts|Country
2 1|Mount Everest/Sagarmatha/Chomolungma|8848|29029|8848|Mahalangur Himalaya|27°59'17"N 86°55'31"E|none|1953|>>145|121|
  Nepal/China
3 2|K2/Qogir/Godwin Austen|8611|28251|4020|Baltoro Karakoram|35°52'53"N 76°30'48"E|Mount Everest|1954|45|44|Pakistan/
  China
4 3|Kangchenjunga|8586|28169|3922|Kangchenjunga Himalaya|27°42'12"N 88°08'51"E|Mount Everest|1955|38|24|Nepal/India
5 4|Lhotse|8516|27940|610|Mahalangur Himalaya|27°57'42"N 86°55'59"E|Mount Everest|1956|26|26|Nepal/China
6 5|Makalu|8485|27838|2378|Mahalangur Himalaya|27°53'23"N 87°05'20"E|Mount Everest|1955|45|52|Nepal/China
7 ...

```

- Sorteer de lijst met bergen eerst numeriek van groot naar klein op absolute hoogte (in meter) en in tweede instantie lexicografisch op naam van de berg. Zorg er evenwel voor dat na sorteren, het eerste veld terug een opeenvolgende nummering vanaf 1 krijgt, ongeacht de oorspronkelijke volgnummers die voor het sorteren gebruikt werden. We krijgen dan (`bergen.4.txt`)

```

1 Rank|Mountain|Height (m)|Height (ft)|Prominence (m)|Range|Coordinates|Parent mountain|First ascent|Ascents|failed
  attempts|Country
2 1|Mount Everest/Sagarmatha/Chomolungma|8848|29029|8848|Mahalangur Himalaya|27°59'17"N 86°55'31"E|none|1953|>>145|121|
  Nepal/China
3 2|Nanga Parbat|8126|26660|4608|Nanga Parbat Himalaya|35°14'14"N 74°35'21"E|Dhaulagiri|1953|52|67|Pakistan
4 3|Jengish Chokusu/Tömür/Pk Pobeda|7439|24406|4148|Tian Shan|42°02'05"N 80°07'47"E|Ismail Samani Peak|1956|||Kyrgyzstan/
  China
5 4|Namcha Barwa|7782|25531|4106|Assam Himalaya|29°37'52"N 95°03'19"E|Kangchenjunga|1992|1|2|China
6 5|K2/Qogir/Godwin Austen|8611|28251|4020|Baltoro Karakoram|35°52'53"N 76°30'48"E|Mount Everest|1954|45|44|Pakistan/
  China
7 ...

```

- Plaats terug komma's tussen de groepen duizendtallen in de drie velden die de hoogte van de berg aangeven. We krijgen dan (`bergen.5.txt`)

```

1 Rank|Mountain|Height (m)|Height (ft)|Prominence (m)|Range|Coordinates|Parent mountain|First ascent|Ascents|failed
  attempts|Country
2 1|Mount Everest/Sagarmatha/Chomolungma|8,848|29,029|8,848|Mahalangur Himalaya|27°59'17"N 86°55'31"E|none
  |1953|>>145|121|Nepal/China
3 2|Nanga Parbat|8,126|26,660|4,608|Nanga Parbat Himalaya|35°14'14"N 74°35'21"E|Dhaulagiri|1953|52|67|Pakistan
4 3|Jengish Chokusu/Tömür/Pk Pobeda|7,439|24,406|4,148|Tian Shan|42°02'05"N 80°07'47"E|Ismail Samani Peak|1956|||
  Kyrgyzstan/China
5 4|Namcha Barwa|7,782|25,531|4,106|Assam Himalaya|29°37'52"N 95°03'19"E|Kangchenjunga|1992|1|2|China
6 5|K2/Qogir/Godwin Austen|8,611|28,251|4,020|Baltoro Karakoram|35°52'53"N 76°30'48"E|Mount Everest|1954|45|44|Pakistan/
  China
7 ...

```

Opgave 6 (repopulator)

(10 pt)

Schrijf een `bash` shell script `repopen` dat kan gebruikt worden om `git repositories` aan te maken, waarbij op een geautomatiseerde manier reeds een aantal `commits` worden uitgevoerd. Aan het shell script moet een bestandsnaam als argument doorgegeven worden. Daarnaast moet het shell script ook de volgende drie opties ondersteunen:

- optie `<int: default 3>`: aantal `commits` met willekeurige `commit messages` (naast de vaste initiële `commit`) die automatisch moeten uitgevoerd worden
- optie `-b <string>`: naam van de `branch` waarin de `commits` met de willekeurige `commit messages` moeten uitgevoerd worden
- optie `-m`: gebruik deze optie om finaal de `branch` waarin de `commits` met de willekeurige `commit messages` werden uitgevoerd te `mergen` met de `master branch`

Het shell script moet voor de verwerking van de opties de flexibiliteit aan de dag leggen die gebruikelijk is bij Unix commando's: volgorde van opties speelt geen rol, opties kunnen eventueel samengenomen worden, Op basis van de gegeven opties en argumenten moet het shell script dan de volgende stappen automatiseren:

1. maak een nieuwe `git repository` aan met de naam die als argument aan het shell script werd doorgegeven
2. voer een initiële `commit` uit (met `commit message initial commit`) in de `master branch` die een leeg bestand (0 bytes) `README.md` aan de `repository` toevoegt
3. voer $c \in \mathbb{N}_0$ `commits` (waarbij c overeenkomt met het verplichte argument dat werd opgegeven bij de optie `-c`, standaard 3) uit in de `branch` waarvan de naam overeenkomt met het verplichte argument dat werd opgegeven bij de optie `-b` (standaard in de `master branch`); hierbij moet de i -de `commit` ($i = 1, 2, \dots$)
 - (a) een willekeurige `commit message` m_i gebruiken die wordt gegenereerd aan de hand van de web service met URL `http://whatthecommit.com/index.txt` (opmerking: als deze web service een `commit message` teruggeeft die bestaat uit meerdere regels, dan moet enkel de eerste regel gebruikt worden)
 - (b) een regel toevoegen aan het bestand `README.md` die bestaat uit het volgnummer i , een afsluitend rond haakje, een spatie en de `commit message` m_i
4. `merge` de `branch` waarin de `commits` in stap 3 uitgevoerd werden met de `master branch`, indien een specifieke `branch` werd opgegeven met de optie `-b` en de optie `-m` werd opgegeven

Geen enkele van de `git` commando's die gebruikt worden om het uitvoeren van bovenstaande stappen te automatiseren mag uitvoer genereren (noch op `stdout` noch op `stderr`). Het shell script moet zelf de volgende foutafhandeling voorzien:

- indien het shell script niet de gepaste opties meekrijgt (enkel ondersteuning voor de opties `-c`, `-b` en `-m`), een optieletter meerdere keren wordt meegegeven, de verplichte argumenten bij de opties worden verkeerd gebruikt, of er wordt niet één enkel argument meegegeven aan het shell script, dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) naar `stderr` uitgeschreven worden, en moet het shell script eindigen met `exit status 1`
- indien aan het shell script een bestandsnaam wordt doorgegeven die reeds bestaat, dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) naar `stderr` uitgeschreven worden, en moet het shell script eindigen met `exit status 2`

Indien er zich geen fouten voordoen, dan moet het shell script eindigen met `exit status 0`.

Voorbeeld

Onderstaande voorbeeldsessie geeft aan hoe het shell script `repogen` moet kunnen gebruikt worden.

```
$ repogen repo01
$ echo $?
0
$ cd repo01; git log --oneline --decorate --graph --all; cd ..
* ffb8bf8 (HEAD -> master) fix bug, for realz
* ce3695b fixed mistaken bug
* ef1c981 For great justice.
* 7ad18b4 initial commit
$ cat repo01/README.md
1) For great justice.
2) fixed mistaken bug
3) fix bug, for realz
$ repogen -c5 repo02
$ cd repo02; git log --oneline --decorate --graph --all; cd ..
* 1b5a2bc (HEAD -> master) added some filthy stuff
* 80ea21f this is how we generate our shit.
* 451835f This is supposed to crash
* e16e601 omgsosorry
* 9fd5801 pay no attention to the man behind the curtain
* 3f52902 initial commit
$ repogen -b develop repo03
$ cd repo03; git log --oneline --decorate --graph --all; cd ..
* 78ad351 (HEAD -> develop) changes
* 2658c9a By works, I meant 'doesnt work'. Works now..
* f844fa4 harharhar
* 309726e (master) initial commit
$ repogen -b develop -m repo04
$ cd repo04; git log --oneline --decorate --graph --all; cd ..
* 68638ea (HEAD -> master, develop) Trust me, I'm an engineer!... What the f*ck did
  just happened here?
* c17b2b4 this doesn't really make things faster, but I tried
* 83a42cf For great justice.
* 0a30c4d initial commit
$ repogen
Syntaxis: repogen [-b <string>] [-c <integer>] [-m] dirname
$ echo $?
1
$ repogen repo01
repogen: bestandsnaam bestaat reeds
$ echo $?
2
```