Computergebruik

# reeks 04: reguliere expressies 24 oktober 2017 22:00

| | Oefening | | Status | |
|---|---|---|---|---|
| ✔ | binair | 96/98 | correct | › |
| ✔ | Populaire voornamen | 92/95 | correct | › |
| ✔ | Baconversleuteling | 68/90 | correct | › |
| ✔ | Kaartspel | 73/86 | correct | › |

## binair

### Opgave

Vanaf pagina 184 in het handboek wordt er uitgelegd hoe reguliere expressies kunnen gebruikt worden om te zoeken in de teksteditor `vi`. In deze opgave wordt het opstellen van dergelijke reguliere expressies ingeoefend. Met behulp van de patroonzoekers `grep`, `egrep` en `fgrep` kun je op een eenvoudige manier bestanden terugvinden waarin een bepaald stukje tekst voorkomt. Daarnaast bieden `egrep` en `grep` ook de mogelijkheid om naar zoekpatronen te zoeken. Een korte samenvatting van de verschillen in functionaliteit kun je vinden in de nota onderaan op pagina 834 in het handboek. Appendix A vanaf pagina 1011 bevat een overzicht over reguliere expressies.

Een zogenaamde **binaire string** is een sequentie van `0`-en en `1`-en, bijvoorbeeld `011011`. De reguliere expressie `[01][01]*` **matcht** dus met elke mogelijke binaire string. Geef nu reguliere expressies die enkel **matchen** met alle binaire strings die:

1. eindigen op `00` (179 regels)
2. beginnen en eindigen met een `1` (0 regels)
3. het patroon `110` bevatten (617 regels)
4. minstens drie `1`-en bevatten (713 regels)
5. minstens drie opeenvolgende `1`-en bevatten (507 regels)

Probeer je expressies opnieuw zo eenvoudig mogelijk te houden. Om lokaal te testen kun je het bestand binair.txt[1] gebruiken.

#### Links

[1]: https://dodona.ugent.be/exercises/1363727109/media/workdir/binair.txt

## Populaire voornamen

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`[1].

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`[2] 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 informatievelden, 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`[3] 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, 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:

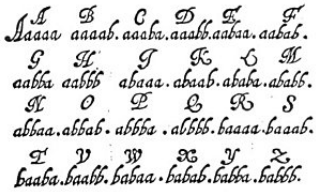
```
7 Robin
6 Robbe
6 Simon
6 Tibo
6 Victor
```

### Links

[1]: https://dodona.ugent.be/exercises/1866903666/media/kieslijst.zip
[2]: https://dodona.ugent.be/exercises/1866903666/media/Q_GE.txt
[3]: https://dodona.ugent.be/exercises/1866903666/media/kieslijst.zip

# Baconversleuteling

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




De sleutel uit Bacons *De Augmentis Scientiarum* (1605) gebruikt het tweeletter-alfabet `a` en `b`.

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*[1] (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).

Het ene lettertype op elke tweede regel staat voor a op de regel erboven, het andere lettertype voor b (uit *De Augmentis Scientiarum*).

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`[2] 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.
   - $\alpha = \{b \in \mathcal{B} \mid$ als je alle letters uit $b$ 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` $\notin \alpha$
   - $\beta = \{b \in \mathcal{B} \mid b$ bevat een reeks van minstens vier opeenvolgende hoofdletters die ingesloten zit tussen dezelfde kleine letter $\}$

voorbeelden: `GnlCyycqHeTilaZoLyjmvLGqpiWHOQTTinm scrimpy` $\in \beta$
`oLRhuiaiabaAllryJZzXGyhQNfnymfvEymgeKAgF maintain` $\notin \beta$

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

  voorbeelden: `FniIfReEpviAZqiuSQiaxxIolYuamFeOhDIoWQEugYAuTxiOQs summerlong` $\in \gamma$
  `ZveSoxrCuqhjwSuCcsUChLpglyHlQNmPgtgHuuVSfBiitmlTitRsmMh sectilities` $\notin \gamma$

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

  voorbeelden: `HsaNWhbCMJvYygvrnyZcdMrKaEwjRbhtFguJfqDO thickset` $\in \delta$
  `ufbvZglmpkQsrUxuFDIrvDXWFdxEGQhYtddaWcTN basophil` $\notin \delta$

Gebruik een commando uit de `grep` familie om enkel die regels van het bestand `bacon.txt`[3] 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 $\alpha$, $\beta$, $\gamma$ en $\delta$ zoals hierboven gedefinieerd. Gebruik nu deze verzamelingen om op de volgende manier een boodschap bestaande uit vier woorden te achterhalen:
   - het eerste woord staat op de unieke regel uit de verzameling $\alpha \cap \beta$
   - het tweede woord staat op de unieke regel uit de verzameling $\beta \cap \gamma$
   - het derde woord staat op de unieke regel uit de verzameling $\gamma \cap \delta$
   - 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.

## Richtlijnen bij het indienen

Volg aandachtig onderstaande richtlijnen bij het indien van je oplossing voor deze opgave:

- Voor deel 2 van de opgave moeten de commando's die je indient enkel het gevraagde woord uitschrijven (zonder de rest van de regel waarop die woorden staan).
- Plaats je commando's voor de acht delen van deze vraag onder de titels in het indienvenster.
- Maak geen aanpassingen aan de regels die al reeds in het venster staan, deze regels worden gebruikt om het bestand op te splitsen in de verschillende deelantwoorden. Op de feedbackpagina kan je controleren of de opsplitsing gelukt is.
- Klik hier om een voorbeeld te vinden van een (foutieve[4]) inzending.

### Links

[1]: https://books.google.be/books?id=l-VBAAAAcAAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
[2]: https://dodona.ugent.be/exercises/576982865/media/workdir/bacon.txt
[3]: https://dodona.ugent.be/exercises/576982865/media/workdir/bacon.txt
[4]: https://dodona.ugent.be/exercises/576982865/media/foutieve_oplossing.txt

# Kaartspel

Een standaard kaartspel bestaat uit 52 *Franse* speelkaarten. Zoals aangegeven in onderstaande tabel zijn ze onderverdeeld in 13 **rangen** (kolommen) voor elk van de 4 **kleuren** (rijen): **klaveren** (♣, *clubs*), **koeken** (♦, *diamonds*), **harten** (♥, *hearts*) en **schoppen** (♠, *spades*). In de tabel zijn de rangen van links naar rechts geordend volgens stijgende waarde.

| | Aas | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Boer | Dame | Heer |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Klaveren** |  | | | | | | | | | | | | |
| **Koeken** |  | | | | | | | | | | | | |
| **Harten** |  | | | | | | | | | | | | |
| **Schoppen** |  | | | | | | | | | | | | |

Een speelkaart wordt genoteerd als twee karakters die respectievelijk de rang en de kleur van de kaart aanduiden. Kleuren worden aangeduid met kleine letters, namelijk de eerste letter van de Engelse benaming voor de kleur (`c`, `d`, `h` en `s`). Rangen worden in oplopende volgorde aangeduid als `A` (**aas**), de cijfers `2` tot en met `9`, `T` (tien), `J` (**boer**, *jack*), `Q` (**dame**, *queen*) en `K` (**heer**, *king*). Klaveren aas wordt dan bijvoorbeeld voorgesteld als `Ac`. De boer, dame of heer worden voorgesteld als symmetrische **prentkaarten** die niet van uitzicht veranderen als je ze ondersteboven bekijkt.

In deze opgave bestaat een **hand** uit 13 verschillende kaarten die willekeurig gekozen werden uit een spel van 52 kaarten. Een hand kan dus voorgesteld worden door 26 karakters, wat meteen ook een volgorde oplegt aan de kaarten in een hand.

## Opgave

Elke regel van het tekstbestand `kaarten.txt`[1] bevat 26 karakters die een hand kaarten voorstellen, gevolgd door één enkele spatie en een woord dat enkel uit letters bestaat. Gevraagd wordt:

1. Bepaal reguliere expressies voor elk van onderstaande verzamelingen. Daarbij staat H voor de verzameling van alle voorstellingen van handen als een reeks van 26 karakters. Probeer de reguliere expressies bovendien zo kort mogelijk te houden.
   - $\alpha = \{h \in H \mid$ h bevat geen prentkaarten$\}$
     voorbeelden: `2s4d5h7cTs6hTh6c5s9d2cTd9s` $\in \alpha$
     `Ah9cJd7d8c5d9hTsAcQc5s4cTc` $\notin \alpha$
   - $\beta = \{h \in H \mid$ h bevat de vier kaarten van eenzelfde rang$\}$
     voorbeelden: `Kd9dJs5sKs7c5c6cKcJhKhTh7h` $\in \beta$
     `AdTdTc2d2cTsKh6c3c6s6dKc4h` $\notin \beta$
   - $\gamma = \{h \in H \mid$ h bevat minstens drie kaarten van elke kleur$\}$
     voorbeelden: `Ad7hTc4h8d8sAsKd5c9cQhJdTs` $\in \gamma$
     `5dKc9cJcTh7sQc3s4sAs7c2cTs` $\notin \gamma$
   - $\delta = \{h \in H \mid$ de kaarten van h zijn gegroepeerd per kleur$\}$ (tussen twee kaarten van eenzelfde kleur staan nooit kaarten van een andere kleur)
     voorbeelden: `9s5s4sKs7h6h4h2d4dAdTd2c3c` $\in \delta$
     `5s9sTs8hKhJc4s6c4hJsAc2dKs` $\notin \delta$

   Gebruik een commando uit de `grep` familie om enkel die regels van het bestand `kaarten.txt`[2] 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.
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:
   - het eerste woord staat op de unieke regel uit de verzameling $\alpha \cap \beta$
   - het tweede woord staat op de unieke regel uit de verzameling $\beta \cap \gamma$
   - het derde woord staat op de unieke regel uit de verzameling $\gamma \cap \delta$
   - het vierde woord staat op de unieke regel uit de verzameling $\delta \cap \alpha$

Links

[1]: https://dodona.ugent.be/exercises/1659479204/media/workdir/kaarten.txt
[2]: https://dodona.ugent.be/exercises/1659479204/media/workdir/kaarten.txt