

---

---

## Computergebruik: EXAMEN

---

---

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

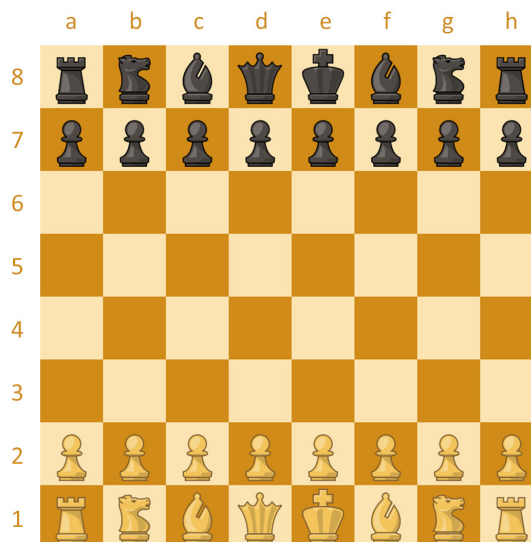
dinsdag 20-08-2019, 14:00  
academiejaar 2018-2019  
tweede zittijd

### Opgave 1 (schaakbord)

(10 pt)

Bij schaken wordt gebruikgemaakt van een **schaakbord**: een vierkant spelbord dat wordt opgedeeld in 32 lichte en 32 donkere velden die elkaar afwisselen. De horizontalen worden **rijen** genoemd en zijn genummerd van 1 tot en met 8. De verticalen worden **lijnen** genoemd en worden aangegeven met de letters a tot en met h. Op die manier kan elk veld aangeduid worden door de combinatie van een letter en een cijfer: de witte koningin staat bijvoorbeeld in de beginstelling op d1. Het bord wordt zo neergelegd dat a1 en h8 — de hoekvelden links van de spelers — zwart zijn.

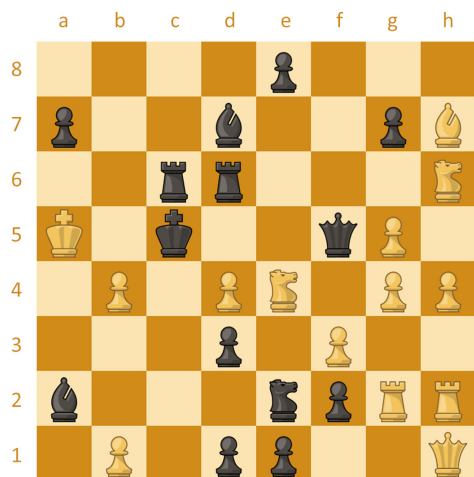
Elke speler beschikt over zestien speelstukken: 1 koning (*king*, K), 1 koningin (*queen*, Q), 2 torens (*rook*, R), 2 lopers (*bishop*, B), 2 paarden (*knight*, N) en 8 pionnen (*pawn*, P). De ene speler heeft witte speelstukken, de andere heeft zwarte speelstukken. Bij aanvang van het spel staan de speelstukken op de volgende manier op het bord.



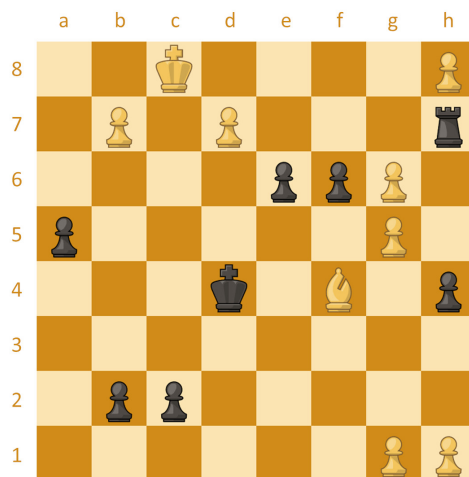
**Forsyth-Edwards Notation** of **FEN** is een standaard voor het beschrijven van de opstelling van de speelstukken op een schaakbord. Het is gebaseerd op een systeem dat in de 19e eeuw werd ontwikkeld door de Schotse journalist David Forsyth, en dat later door Steven J Edwards werd uitgebreid zodat het kon gebruikt worden in computerprogramma's. Zo wordt bijvoorbeeld de beginopstelling van de speelstukken in FEN genoteerd als

```
rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR
```

Hierbij wordt elke rij (gezien vanuit het standpunt van de witte speler) beschreven, beginnend met de achtste rij en eindigend met de eerste rij. De rijen worden gescheiden door een slash (/). Binnen elke rij wordt de inhoud van elk veld beschreven van de a-lijn tot de h-lijn. Witte stukken worden aangegeven met hun Engelse aanduiding in hoofdletters (KQRBNP), zwarte stukken met diezelfde aanduiding in kleine letters (kqrbnp). Het aantal opeenvolgende lege velden op een rij wordt aangegeven met een cijfer dat van 1 tot en met 8 kan lopen. Hieronder zie je nog twee voorbeelden van speelstukken op een schaakbord met bijhorende omschrijving in FEN-notatie.



4p3/p2b2pB/2rr3N/K1k2qP1/1P1PN1PP/3p1P2/b3npRR/1P1pp2Q



2K4P/1P1P3r/4ppP1/p5P1/3k1B1p/8/1pp5/6PP

## Opgave

Elke regel van het tekstbestand `FEN.txt` bestaat uit een patroon  $p \in \mathcal{P}$ , gevolgd door een spatie en een woord  $w \in \mathcal{W}$ . De verzameling  $\mathcal{P}$  bevat de FEN-notaties van alle mogelijke manieren waarop een deel van de speelstukken op een schaakbord kunnen geplaatst worden. De verzameling  $\mathcal{W}$  bevat alle woorden die enkel bestaan uit letters. Gevraagd wordt:

1. Bepaal zo kort mogelijke reguliere expressies voor de volgende deelverzamelingen van  $\mathcal{P}$ :

(a)  $\mathcal{P}_1 = \{p \in \mathcal{P} \mid \text{er staan 8, 9, 10, 11 of 12 pionnen op het schaakbord}\}$

voorbeeld: `1p3qp1/R2P2Pp/2n5/1pP5/kbP2p2/1P4B1/5K2/2B1Q2N`  $\in \mathcal{P}_1$   
`7R/1p6/2b1B1P1/3npp2/1pn5/Q3PP2/1r5K/4q1R1`  $\notin \mathcal{P}_1$

(b)  $\mathcal{P}_2 = \{p \in \mathcal{P} \mid \text{er staat minstens één pion op elke rij van het schaakbord}\}$

voorbeeld: `ppQp1K2/2Pb4/3k2nP/5P2/1p5B/P3P1NR/2p1PpNP/7p`  $\in \mathcal{P}_2$   
`6pP/3p2P1/rP1n4/8/2Bp1P1b/8/3P1n2/K3Nr2`  $\notin \mathcal{P}_2$

(c)  $\mathcal{P}_3 = \{p \in \mathcal{P} \mid \text{op het schaakbord staan geen twee speelstukken van dezelfde soort maar met een verschillende kleur naast elkaar op dezelfde rij}\}$

voorbeeld: `r1b2pp1/pq1r3R/8/P1b5/3p3Q/p1Pn4/PNN2R1p/1PB2P2`  $\in \mathcal{P}_3$   
`1K5p/7P/3p3P/1P1p3p/5R1P/1np3p1/1k1Rr1B1/5Q2`  $\notin \mathcal{P}_3$

(d)  $\mathcal{P}_4 = \{p \in \mathcal{P} \mid \text{er staat een zwart paard op de buitenste rand van het schaakbord}\}$

voorbeeld: `1P2P2B/6pn/1PP2bqp/1r5p/r2pPP1p/1R1NP3/3B4/Nn4p1`  $\in \mathcal{P}_4$   
`QR4p1/1Pp5/3p2B1/N1n4b/2p2Nr1/1n5P/1P3B2/3p2RK`  $\notin \mathcal{P}_4$

Geef telkens een Unix commando waarin de reguliere expressie gebruikt wordt door een commando uit de `grep` familie om enkel de regels van het tekstbestand naar `stdout` te schrijven waarvan het patroon  $p$  behoort tot  $\mathcal{P}_i$  ( $i = 1, 2, 3, 4$ ).

2. Bepaal als volgt de woorden  $w_1$   $w_2$   $w_3$   $w_4$  van een geheime boodschap:

(a) het woord  $w_1$  staat op de unieke regel waarvan  $p$  behoort tot  $\mathcal{P}_1 \cap \mathcal{P}_2$

(b) het woord  $w_2$  staat op de unieke regel waarvan  $p$  behoort tot  $\mathcal{P}_2 \cap \mathcal{P}_3$

(c) het woord  $w_3$  staat op de unieke regel waarvan  $p$  behoort tot  $\mathcal{P}_3 \cap \mathcal{P}_4$

(d) het woord  $w_4$  staat op de unieke regel waarvan  $p$  behoort tot  $\mathcal{P}_4 \cap \mathcal{P}_1$

Geef telkens een Unix commando waarin de reguliere expressies voor de verzamelingen  $\mathcal{P}_i$  ( $i = 1, 2, 3, 4$ ) gebruikt worden door commando's uit de **grep** familie om het woord  $w_j$  ( $j = 1, 2, 3, 4$ ) op te zoeken in het tekstbestand en uit te schrijven naar **stdout**. Hierbij is het niet toegelaten om het woord  $w_j$  letterlijk uit te schrijven (bv. **echo**  $w_j$ ).

## Opgave 2 (Unicode-lettertypes)

(10 pt)

De **Unicode-standaard** specificeert geen lettertype — een verzameling van grafische vormen die gliefen genoemd worden. In plaats daarvan worden abstracte tekens geïdentificeerd door een uniek nummer (een zogenaamd codepunt), en worden ook de vormveranderingen vastgelegd die moeten gebeuren afhankelijk van de context waarin een glief gebruikt wordt (bv. het combineren van tekens, vooraf gecombineerde tekens, en vaste combinaties van letters en diakritische tekens). De keuze van het lettertype — dat bepaalt hoe de abstracte tekens van Unicode moeten omgezet worden naar een bitmap- of vectorvoorstelling die op een scherm kan weergegeven worden of afgedrukt kan worden — wordt dan aan de gebruiker overgelaten. Als een lettertype gekozen wordt dat geen glief bevat voor een bepaald codepunt dat in een document gebruikt wordt, dan wordt de glief doorgaans weergegeven als een vraagteken, een vierkantje, of een ander plaatsvervangend teken.

Computerlettertypes gebruiken verschillende technieken om tekens of gliefen weer te geven. Bitmap-lettertypes gebruiken een raster van stippen — pixels genoemd — die van elke glief een afbeelding vormen in elke verschijningsvorm en in elke grootte. Omtreklettertypes (ook vectorlettertypes genoemd) gebruiken tekeninstructies of wiskundige formules om elke glief te beschrijven. Lijnlettertypes gebruiken een reeks van gespecificeerde lijnen en aanvullende informatie om het profiel te definiëren — de dikte en de vorm van de lijn in een specifieke verschijningsvorm en grootte — die dan samen de weergave van de glief beschrijven.

Lettertypes bevatten ook speciale orthografische regels om meerdere gliefen met elkaar te kunnen combineren tot ligaturen (gecombineerde tekens). Besturingssystemen, webbrowsers en andere software die gebruikmaakt van typografie, gebruiken lettertypes om tekst op het scherm of afdrukmedia weer te geven, en kunnen geprogrammeerd worden om die orthografische regels toe te passen. Als alternatief kunnen ze externe scripts gebruiken om de vorm van gliefen aan te passen (rendertechnologie of "smart font"-engines), en ze kunnen ook geprogrammeerd worden om ofwel één groot Unicode-lettertype te gebruiken of om meerdere lettertypes te combineren voor verschillende tekens of talen.

Geen enkel Unicode-lettertype definieert alle tekens uit de meest recente versies van de Unicode-standaard, omdat er steeds meer talen en tekens aan toegevoegd worden, en omdat veelgebruikte lettertypeformaten niet meer dan 65.535 gliefen kunnen bevatten. Dat is ongeveer de helft van de 137.994 tekens uit Unicode 12.1 — de meest recente versie uitgebracht in mei 2019 — die naast 150 moderne en historische schriften ook verschillende collecties symbolen en emoji's bevat. De eerste Unicode-lettertypes met zeer grote tekensets en ondersteuning voor veel Unicode-blokken waren Lucida Sans Unicode (1993), Unihan font (1993) en Everson Mono (1995).

### Opgave

Een **Unicode-lettertype** is een computerlettertype dat Unicode-tekens afbeeldt op gliefen. Daardoor kunnen de gliefen van het lettertype geïdentificeerd worden door codepunten zoals gedefinieerd in de Unicode-standaard. De overgrote meerderheid van de moderne computerlettertypes is gebaseerd op Unicode, zelfs die lettertypes die enkel gliefen bevatten voor één enkel schrift, of die zelfs enkel het standaard Latijnse alfabet ondersteunen. Lettertypes die een breed scala aan Unicode-schriften en Unicode-symbolen ondersteunen, worden "pan-Unicode lettertypes" genoemd. Maar omdat het maximale aantal gliefen dat kan gedefinieerd worden in een TrueType-lettertype beperkt is tot 65.535, kan één enkel lettertype onmogelijk alle individuele gliefen bevatten voor alle Unicode-tekens.

Het tekstbestand `unicode.txt` bevat een lijst van veelgebruikte Unicode-lettertypes die een relatief groot aantal en een breed scala aan Unicode-tekens ondersteunen. Elke regel van het bestand be-

vat informatie over één lettertype, bestaande uit de volgende twaalf velden die van elkaar worden gescheiden door tabs: *i*) naam (Font), *ii*) licentie (License), *iii*) aantal tekens (Chars), *iv*) aantal gliefen (Glyphs), *v*) meest recente versie en publicatiedatum (Version (date)), *vi*) naam en grootte van het bestand (Filename (size)), *vii*) lettertypefamilie (Font family), *viii*) verschijningsvormen (Font weight, style), *ix*) soort lettertype (Font type), *x*) soorten schreef (Serif style), *xi*) ontwerper/auteur en auteursrecht (Creator/author (copyright)), en *xii*) opmerkingen (Notes). Je mag ervan uitgaan dat de velden zelf geen tabs en puntkomma's (;) bevatten.

Gevraagd wordt om — gebruikmakend 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. De opdrachten mogen de eerste regel niet wijzigen, tenzij dit expliciet anders vermeld staat. Ter controle kan je gebruikmaken van de meegeleverde bestanden `unicode.i.txt` (ZIP), die telkens de inhoud van het bestand bevatten nadat de *i*-de opdracht werd uitgevoerd.

### 1. Verwijder de laatste twee velden, ook op de eerste regel. Zo moet

Font	License	Chars	Glyphs	Version (date)	Filename (size)	Font family	Font weight, style	Font type	Serif style	Creator/author (copyright)	Notes
Arial	Proprietary	3988	3988	v6.80 (2012-05-11)	arial.ttf (756 KB)	Arial	Medium (Normal), Regular	OTF+TTO Normal Sans	(2008) Monotype Corporation	Included with Microsoft Windows	
Arial Unicode MS	Proprietary	38917	50377	v1.01 sfont rev 1 (2002-11-18)	ARIALUNI.TTF (22.1 MB)	ARIALUNI.TTF	Medium (Normal), Regular	OTF+TTO Normal Sans	(2008) Monotype Corporation	Included with Microsoft Windows	
Bitstream Cyberbit	Freeware	32961	29934	v2.0 beta (1998-03-22)	Cyberbit.ttf (12.7 MB)	Bitstream Cyberbit	Medium (Normal), Roman	TTF	Cove	Bitstream Inc. For non-commercial use only.	
Bitstream CyberCJK	Freeware	30275	28686	v2.0 beta (1998-03-17)	Cyberbit.ttf (12.4 MB)	Bitstream CyberCJK	Medium (Normal), Book, Roman	TTF	Cove	Bitstream Inc. For non-commercial use only.	
Brampton	OFL	1916	1979	v0.16 (2011-09-22)	Brampton.ttf (622 KB)	Brampton	TTF	Slab serif	Christ Trek fonts		

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

Font	License	Chars	Glyphs	Version (date)	Filename (size)	Font family	Font weight, style	Font type	Serif style
Arial	Proprietary	3988	3988	v6.80 (2012-05-11)	arial.ttf (756 KB)	Arial	Medium (Normal), Regular	OTF+TTO Normal Sans	
Arial Unicode MS	Proprietary	38917	50377	v1.01 sfont rev 1 (2002-11-18)	ARIALUNI.TTF (22.1 MB)	ARIALUNI.TTF	Medium (Normal), Regular	OTF+TTO Normal Sans	
Bitstream Cyberbit	Freeware	32961	29934	v2.0 beta (1998-03-22)	Cyberbit.ttf (12.7 MB)	Bitstream Cyberbit	Medium (Normal), Roman	TTF	Cove
Bitstream CyberCJK	Freeware	30275	28686	v2.0 beta (1998-03-17)	Cyberbit.ttf (12.4 MB)	Bitstream CyberCJK	Medium (Normal), Book, Roman	TTF	Cove
Brampton	OFL	1916	1979	v0.16 (2011-09-22)	Brampton.ttf (622 KB)	Brampton	TTF	Slab serif	

### 2. Vervang de tabs als veldscheidingsteken door puntkomma's (;), ook op de eerste regel. We krijgen dan (`unicode.2.txt`)

Font;License;Chars;Glyphs;Version (date);Filename (size);Font family;Font weight, style;Font type;Serif style
Arial;Proprietary;3988;3988;v6.80 (2012-05-11);arial.ttf (756 KB);Arial;Medium (Normal), Regular;OTF+TTO;Normal Sans
Arial Unicode MS;Proprietary;38917;50377;v1.01 sfont rev 1 (2002-11-18);ARIALUNI.TTF (22.1 MB);ARIALUNI.TTF;Medium (Normal), Regular;OTF+TTO;Normal Sans
Bitstream Cyberbit;Freeware;32961;29934;v2.0 beta (1998-03-22);Cyberbit.ttf (12.7 MB);Bitstream Cyberbit;Medium (Normal), Roman;TTF;Cove
Bitstream CyberCJK;Freeware;30275;28686;v2.0 beta (1998-03-17);Cyberbit.ttf (12.4 MB);Bitstream CyberCJK;Medium (Normal), Book, Roman;TTF;Cove
Brampton;OFL;1916;1979;v0.16 (2011-09-22);Brampton.ttf (622 KB);Brampton;TTF;Slab serif

### 3. Verwijder alle gepatenteerde lettertypes. Dat zijn de lettertypes waarvan de inhoud van het licentieveld (License) gelijk is aan Proprietary. We krijgen dan (`unicode.3.txt`)

Font;License;Chars;Glyphs;Version (date);Filename (size);Font family;Font weight, style;Font type;Serif style
Bitstream Cyberbit;Freeware;32961;29934;v2.0 beta (1998-03-22);Cyberbit.ttf (12.7 MB);Bitstream Cyberbit;Medium (Normal), Roman;TTF;Cove
Bitstream CyberCJK;Freeware;30275;28686;v2.0 beta (1998-03-17);Cyberbit.ttf (12.4 MB);Bitstream CyberCJK;Medium (Normal), Book, Roman;TTF;Cove
Brampton;OFL;1916;1979;v0.16 (2011-09-22);Brampton.ttf (622 KB);Brampton;TTF;Slab serif
Cardo;OFL;2879;2882;v0.099 sfont rev 0.098999 (2010-05-23);Cardo999.ttf (706 KB);Cardo;Medium (Normal), Book, Regular;TTF;Cove
Carlson Roman;BSD-like license;3683;3686;v001.000 (2003-10-23);CarlsonRoman.sfd (3.70 MB);Carlson;Roman;TTF

### 4. Rangschik de lettertypes lexicografisch volgens het licentieveld (License), en rangschik lettertypes met dezelfde licentie volgens dalend aantal karakters (Chars). We krijgen dan (`unicode.4.txt`)

Font;License;Chars;Glyphs;Version (date);Filename (size);Font family;Font weight, style;Font type;Serif style
Carlson Roman;BSD-like license;3683;3686;v001.000 (2003-10-23);CarlsonRoman.sfd (3.70 MB);Carlson;Roman;TTF
DejaVu Sans;Bitstream Vera license and public domain for additions;5467;5762;v2.32 sfont rev 2.31999 (2010-08-22);DejaVuSans.ttf (667 KB);DejaVu Sans;Medium (Normal), Book;OTF+TTO;Normal Sans
Bitstream Cyberbit;Freeware;32961;29934;v2.0 beta (1998-03-22);Cyberbit.ttf (12.7 MB);Bitstream Cyberbit;Medium (Normal), Roman;TTF;Cove
Bitstream CyberCJK;Freeware;30275;28686;v2.0 beta (1998-03-17);Cyberbit.ttf (12.4 MB);Bitstream CyberCJK;Medium (Normal), Book, Roman;TTF;Cove
YdFontN;Freeware;21957;57621;v13.00 sfont rev 5 Pen-Ji (2010-08-24);YdFontN.TTC (13.5 MB);YdFontN;Regular;TTC;Any

### 5. Gebruik een komma (,) om de duizendtallen te scheiden in de velden met het aantal tekens (Chars) en het aantal gliefen (Glyphs). We krijgen dan (`unicode.5.txt`)

Font;License;Chars;Glyphs;Version (date);Filename (size);Font family;Font weight, style;Font type;Serif style
Carlson Roman;BSD-like license;3,683;3,686;v001.000 (2003-10-23);CarlsonRoman.sfd (3.70 MB);Carlson;Roman;TTF
DejaVu Sans;Bitstream Vera license and public domain for additions;5,467;5,762;v2.32 sfont rev 2.31999 (2010-08-22);DejaVuSans.ttf (667 KB);DejaVu Sans;Medium (Normal), Book;OTF+TTO;Normal Sans
Bitstream Cyberbit;Freeware;32,961;29,934;v2.0 beta (1998-03-22);Cyberbit.ttf (12.7 MB);Bitstream Cyberbit;Medium (Normal), Roman;TTF;Cove
Bitstream CyberCJK;Freeware;30,275;28,686;v2.0 beta (1998-03-17);Cyberbit.ttf (12.4 MB);Bitstream CyberCJK;Medium (Normal), Book, Roman;TTF;Cove
YdFontN;Freeware;21,957;57,621;v13.00 sfont rev 5 Pen-Ji (2010-08-24);YdFontN.TTC (13.5 MB);YdFontN;Regular;TTC;Any

6. Het veld met de meest recente versie en de publicatiedatum (**Version (date)**), en het veld met de naam en de grootte van het bestand (**Filename (size)**) hebben beide een inhoud van de vorm **deel1 (deel2)**. Splits beide velden in twee velden (**deel1** en **deel2**), zodat deze nieuwe velden ook van elkaar gescheiden worden door puntkomma's. Doe dit ook met de veldnamen op de eerste regel. We krijgen dan (**unicode.6.txt**)

```
1 Font;License;Chars;Glyphs;Version;date;Filename;size;Font family;Font weight , style;Font type;Serif style
2 Caslon Roman;BSD-like license;3,683;3,686;v001.000;2003-10-23;CaslonRoman.sfd;3.70 MB;Caslon;Roman;TTF
3 DejaVu Sans;Bitstream Vera license and public domain for additions;5,467;5,762;v2.32 sfont rev 2.31999;2010-08-22;DejaVuSans.ttf;667 KB;DejaVu Sans;Medium (Normal), Book;OTF+TTO;Normal Sans
4 Bitstream Cyberbit;FreeWare;32,961;29,934;v2.0 beta;1998-03-22;Cyberbit.ttf;12.7 MB;Bitstream Cyberbit;Medium (Normal), Roman;TTF;Cove
5 BitstreamCyberCJK;FreeWare;30,275;28,686;v2.0 beta;1998-03-17;Cyberbit.ttf;12.4 MB;Bitstream CyberCJK;Medium (Normal), Book, Roman;TTF;Cove
6 Y.OzFontN;FreeWare;21,967;57,621;v13.00 sfont rev 5 Pen-Ji;2010-08-24;YozRN.TTC;13.5 MB;YozFontN;Regular;TTC;Any
7 ...
```

7. Zet alle veldeninhouden tussen dubbele aanhalingstekens, inclusief de veldnamen op de eerste regel. We krijgen dan (**unicode.7.txt**)

```
1 "Font";"License";"Chars";"Glyphs";"Version";"date";"Filename";"size";"Font family";"Font weight , style";"Font type";"Serif style"
2 "Caslon Roman";"BSD-like license";"3,683";"3,686";"v001.000";"2003-10-23";"CaslonRoman.sfd";"3.70 MB";"Caslon";"Roman";"TTF";
3 "DejaVu Sans";"Bitstream Vera license and public domain for additions";"5,467";"5,762";"v2.32 sfont rev 2.31999";"2010-08-22";"DejaVuSans.ttf";"667 KB";"DejaVu Sans";"Medium (Normal), Book";"OTF+TTO";"Normal Sans"
4 "Bitstream Cyberbit";"FreeWare";"32,961";"29,934";"v2.0 beta";"1998-03-22";"Cyberbit.ttf";"12.7 MB";"Bitstream Cyberbit";"Medium (Normal), Roman";"TTF";"Cove"
5 "BitstreamCyberCJK";"FreeWare";"30,275";"28,686";"v2.0 beta";"1998-03-17";"Cyberbit.ttf";"12.4 MB";"Bitstream CyberCJK";"Medium (Normal), Book, Roman";"TTF";"Cove"
6 "Y.OzFontN";"FreeWare";"21,967";"57,621";"v13.00 sfont rev 5 Pen-Ji";"2010-08-24";"YozRN.TTC";"13.5 MB";"YozFontN";"Regular";"TTC";"Any"
7 ...
```

### Opgave 3 (verhaallijn)

(10 pt)

We hebben een bekende **oneliner** uit een film verborgen in een directorystructuur. Jouw missie — mocht je kiezen om die te aanvaarden — bestaat erin om die oneliner op te sporen. Om uit te leggen hoe we de oneliner in de directorystructuur verborgen hebben, gebruiken we het volgende voorbeeld:

I see dead people.

Als leidraad gebruiken we een tekstbestand met daarin de korte **verhaallijn** van de film. In dit voorbeeld gebruiken we het tekstbestand **the\_sixth\_sense.txt** met daarin de volgende verhaallijn van de film *The Sixth Sense* (1999):

"A boy who communicates with spirits, seeks the help of a disheartened child-psychologist."

We herleiden deze verhaallijn tot een opeenvolging van woorden, waarbij een woord gedefinieerd wordt als de langst mogelijke opeenvolging van letters. De woorden worden daarbij ook omgezet naar kleine letters. Bovenstaande verhaallijn wordt dus herleid tot de volgende reeks van woorden:

a boy who communicates with spirits seeks the help of a disheartened child psychologist

We maken nu een directory (bv. **the\_sixth\_sense/**) die we de **basisdirectory** noemen, en navigeren naar de basisdirectory. Daarna doen we met de  $n$  opeenvolgende woorden  $w_i$  ( $i = 1, \dots, n$ ) uit de reeks telkens willekeurig één van de volgende twee dingen:

- we maken een subdirectory  $w_i$  in de huidige directory en navigeren naar die subdirectory
- we maken een gewoon bestand  $w_i$  in de huidige directory met als inhoud het volgende woord van de oneliner; daarna navigeren we terug naar de basisdirectory

Op die manier verkrijgen we bijvoorbeeld onderstaande directorystructuur:

```

the_sixth_sense/
├── a/
│   ├── boy ("I")
│   └── disheartened/
│       └── child/
│           └── psychologist ("people.")
├── who/
│   └── communicates ("see")
└── with/
    ├── spirits/
    │   └── seeks/
    │       └── the/
    │           └── help/
    │               └── of ("dead")

```

Daarbij hebben we de directories voorgesteld met een slash op het einde van hun naam, en hebben we na de namen van de gewone bestanden tussen ronde haakjes en dubbele aanhalingstekens ook hun inhoud gezet. Om de oneliner nog beter te verbergen, voegen we daarna onder de basisdirectory ook nog wat willekeurige directories en gewone bestanden toe. Het resultaat hiervan vind je terug in het ZIP-bestand `the_sixth_sense.zip`.

## Opgave

Schrijf een `bash` shell script `oneliner` waaraan twee argumenten moeten doorgegeven worden: *i*) de padnaam van een basisdirectory waarin we een oneliner uit een film verborgen hebben volgens de procedure uit de inleiding en *ii*) de padnaam van een tekstbestand met de korte verhaallijn van een film die we hierbij als leidraad gebruikt hebben. Het shell script moet de verborgen oneliner uitschrijven naar `stdout`, waarbij de woorden van de oneliner telkens van elkaar worden gescheiden door één enkele spatie.

## Voorbeeld

Onderstaande voorbeeldsessie geeft aan hoe het shell script `oneliner` moet kunnen gebruikt worden. Hierbij gaan we ervan uit dat de directory `the_sixth_sense` en het bestand `the_sixth_sense.txt` zich in de huidige directory bevinden.

```

$ oneliner the_sixth_sense the_sixth_sense.txt
I see dead people.

```

## Opgave 4 (verjaardagen)

(20 pt)

Je legt een bestand aan met geboortedatums van jouw vrienden, zodat je makkelijker hun verjaardagen en leeftijden kan onthouden. Elke regel van het bestand bevat de volgende vier informatievelen die van elkaar gescheiden worden door spaties: *i*) geboortejaar, *ii*) geboortemaand, *iii*) geboortedag en *iv*) naam. Het veld met de naam mag zelf ook spaties bevatten. Het bestand zou er bijvoorbeeld als volgt kunnen uitzien:

```

1912 6 23 Alan Turing
1955 6 8 Tim Berners-Lee
1938 1 10 Donald Knuth
1973 8 12 Sergey Brin
1973 3 26 Larry Page
1943 2 4 Ken Thompson

```

## Opgave

Schrijf een `bash` shell script `birthdays` dat een overzicht uitschrijft naar `stdout` van je vrienden die de komende periode jarig zijn. Elke regel van het overzicht bestaat uit de naam van een persoon, een spatie en tussen ronde haakjes wanneer de persoon de volgende keer jarig is. Hiervoor baseert het script zich op een bestand waarvan de inhoud in de inleiding werd beschreven.

Het script moet de volgende opties ondersteunen:

- optie `-a`: schrijf tussen de ronde haakjes (vóór de verjaardag en ervan gescheiden door een komma en een spatie) ook de leeftijd  $n$  op de volgende verjaardag uit in het formaat `age n`
- optie `-c <int>`: schrijf alle personen uit die de komende `<int>` dagen jarig zijn; dit is tot en met `<int>` dagen na vandaag en inclusief vandaag; standaard worden alle personen uitgeschreven die de komende 7 dagen jarig zijn
- optie `-d <file>`: gebruik de personen en hun geboortedatums uit het bestand op locatie `<file>`; standaard maakt het script gebruik van het bestand `~/birthdays`
- optie `-h`: verjaardagen die vandaag vallen worden uitgeschreven als `today`, verjaardagen die morgen vallen als `tomorrow`, en verjaardagen die over  $n > 1$  dagen vallen als `in n days`; standaard worden verjaardagen uitgeschreven in het formaat `jaar/maand/dag` (zoals ze in het bestand voorkomen)
- optie `-s`: sorteer de personen volgens toenemend aantal dagen tot de volgende verjaardag; standaard worden de personen uitgeschreven in de volgorde waarin ze in het bestand voorkomen

Het script moet bij 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, .... Daarnaast moet het script ook de volgende foutafhandeling voorzien:

- als het script niet de gepaste opties meekrijgt (enkel ondersteuning voor de hierboven opgelijste opties) dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stderr` en moet het script eindigen met `exit status 1`
- als het bestand niet gevonden wordt op de aangegeven locatie (of de standaardlocatie indien er niet expliciet een locatie wordt aangegeven) of daar niet kan uitgelezen worden, dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stderr` en moet het script eindigen met `exit status 2`

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

### Tip

Om het verschil tussen twee datums te bepalen, kan je de datums omzetten naar Unix-tijd (aantal seconden sinds een referentietijdstip, meestal 1 januari 1970 om 00:00:00 UTC):

```
$ date -d "19 aug 2019" "+%s"
1566165600
```

## Voorbeeld

Onderstaande voorbeeldsessie geeft aan hoe het `bash` shell script `birthdays` moet kunnen gebruikt worden. Hierbij gaan we ervan uit dat het tekstbestand `.birthdays` zich in je home directory bevindt en dat het tekstbestand `scientists.txt` zich in de huidige directory bevindt. Het voorbeeld gaat er ook van uit dat het bestand `unknown` niet bestaat (of niet leesbaar is).

```
$ date
Mon Aug 19 15:22:54 DST 2019
$ birthdays
Sergey Brin (21/8/2019)
Masatoshi Shima (22/8/2019)h
Jeff Rulifson (20/8/2019)
Edgar F. Codd (19/8/2019)
$ birthdays -c 30
Sergey Brin (21/8/2019)
Dennis Ritchie (9/9/2019)
Masatoshi Shima (22/8/2019)
John McCarthy (4/9/2019)
Jeff Rulifson (20/8/2019)
Edgar F. Codd (19/8/2019)
$ birthdays -c30 -a
Sergey Brin (age 46, 21/8/2019)
Dennis Ritchie (age 78, 9/9/2019)
Masatoshi Shima (age 76, 22/8/2019)
John McCarthy (age 93, 4/9/2019)
Jeff Rulifson (age 78, 20/8/2019)
Edgar F. Codd (age 96, 19/8/2019)
$ birthdays -c30 -ah
Sergey Brin (age 46, in 2 days)
Dennis Ritchie (age 78, in 21 days)
Masatoshi Shima (age 76, in 3 days)
John McCarthy (age 93, in 16 days)
Jeff Rulifson (age 78, tomorrow)
Edgar F. Codd (age 96, today)
$ birthdays -c30 -ahs
Edgar F. Codd (age 96, today)
Jeff Rulifson (age 78, tomorrow)
Sergey Brin (age 46, in 2 days)
Masatoshi Shima (age 76, in 3 days)
John McCarthy (age 93, in 16 days)
Dennis Ritchie (age 78, in 21 days)
$ birthdays -a -c 14 -d scientists.txt -h -s
Antoine-Laurent de Lavoisier (age 276, in 7 days)
Ernest Rutherford, 1st Baron Rutherford of Nelson (age 148, in 11 days)
$ birthdays -x
birthdays [-ahs] [-c <int>] [-d <file>]
$ echo $?
1
$ birthdays -d
birthdays [-ahs] [-c <int>] [-d <file>]
$ echo $?
1
$ birthdays -d unknown
birthdays: cannot access database 'unknown'
$ echo $?
2
```