

---

---

## EXAMEN: Scriptingtalen

---

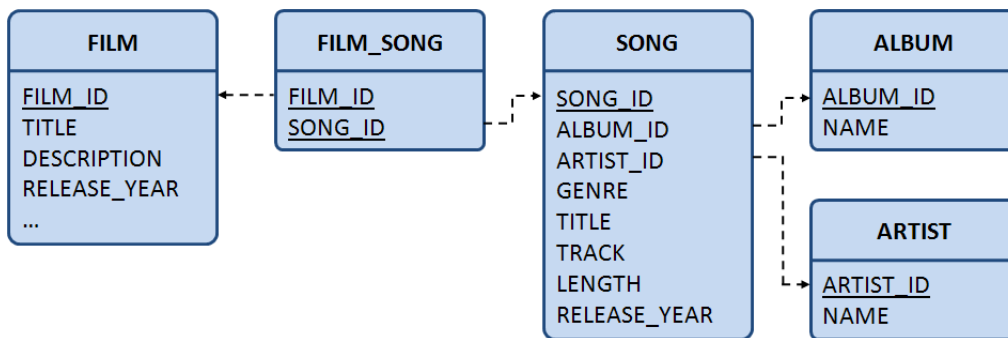
---

Prof. Dr. Peter Dawyndt  
1<sup>e</sup> Bachelor Informatica  
groep 1-2

maandag 23-06-2008, 8:30h  
academiejaar 2007-2008  
eerste zittijd

### Opgave 1

De DVD-verhuurketen heeft zijn Sakila databank uitgebreid, waardoor informatie kan worden opgezocht over soundtracks van films. Hiervoor werden vier extra tabellen aan de oorspronkelijke databank toegevoegd. Onderstaande figuur geeft een overzicht van de nieuwe tabellen, waarbij (samengestelde) primaire sleutels onderlijnd zijn en vreemde sleutels worden aangegeven met onderbroken pijlen.



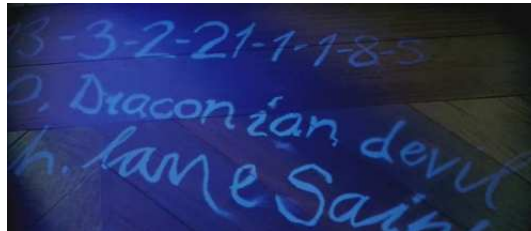
De tabel **SONG** houdt een lijst bij van liedjes, waarbij aan elk liedje een uniek rangnummer toegekend wordt (kolom **SONG\_ID**). De kolom **ALBUM\_ID** is een vreemde sleutel die verwijst naar de gelijknamige kolom van de tabel **ALBUM**, en die aangeeft op welk album het liedje voorkomt. De kolom **ARTIST\_ID** is een vreemde sleutel die verwijst naar de gelijknamige kolom van de tabel **ARTIST**, en die aangeeft welke artiest het nummer geschreven heeft. De tabel **ALBUM** (resp. **ARTIST**) bevat verder enkel de naam van het album (resp. de artiest). Voorts geeft de tabel **SONG** ook nog informatie weer over de titel van het nummer (**TITLE**), het genre (**GENRE**), de plaats van het nummer op het album (**TRACK**), de lengte van het nummer (**LENGTH**) en het jaar waarin het werd uitgegeven (**RELEASE\_YEAR**). De tabel **FILM\_SONG** geeft aan welk nummer in welke film gebruikt wordt, en bevat daarvoor twee vreemde sleutels die verwijzen naar de primaire sleutels van de tabellen **FILM** en **SONG**.

De uitgebreide Sakila databank werd ingeladen in een MySQL RDBMS en is bevroegbaar via de phpMyAdmin webapplicatie (<http://users.ugent.be/~dhollevo/scrita.sql/>). Inloggen doe je in deze databank met je UGent account. Gevraagd wordt om SQL zoekopdrachten te formuleren die een antwoord bieden op onderstaande vragen. Zorg er telkens voor dat de kolommen van de resulterende tabel een zinvolle naam krijgen. Geef in je antwoordbestand ook aan hoeveel records de resultatentabel bevat.

1. Bereken het minimum, maximum en gemiddeld aantal liedjes per film.
2. Uit welk jaar zijn de meeste liedjes afkomstig? Geef ook het aantal liedjes dat in dit jaar werd geschreven.
3. Geef een alfabetisch gerangschikte lijst van de titels van alle films waarin muziek gebruikt wordt uit het album "The Ideal Crash".
4. Geef de namen van elk paar artiesten dat aan minstens dertig gemeenschappelijke films een soundtrack heeft geleverd. Sorteert elk paar eerst op de eerste naam en dan op de tweede naam.

## Opgave 2

Een anagram is een woord dat of een zin die bestaat uit alle letters van een ander woord of een andere zin. Anagrammen worden vaak gebruikt als pseudoniem (*Leonardo da Vinci: o draconian devil; The Mona Lisa: oh lame saint*). Een eenvoudige manier om te bepalen of woorden anagrammen zijn, bestaat erin om een hash-waarde te berekenen zodat enkel anagrammen identieke waarden hebben. Als mogelijke hash-waarde kan je bijvoorbeeld gebruik maken van de alfabetisch gesorteerde letters van het originele woord. Bij de berekening van de hash-waarde moeten enkel de letters uit het alfabet in rekening gebracht worden (geen leestekens, cijfers, etc.), en moeten deze niet-hoofdlettergevoelig behandeld worden. Zo resulteren de woorden *callers*, *cellars* en *RECALLS* allemaal in de hash-waarde *acellrs*, waardoor het dus anagrammen zijn.



Gevraagd wordt om een **awk**-script `anagram.awk` te schrijven dat alle combinaties van woorden die anagrammen zijn opzoekt in een gegeven woordenlijst. Het script leest de woordenlijst uit een bestand dat één enkel woord per regel bevat. Elke regel die door het script naar standaard uitvoer wordt weggeschreven moet alle woorden uit de woordenlijst bevatten die anagrammen van elkaar zijn. De anagrammen worden opeend genummerd, waarbij voor elke index vier posities gereserveerd worden en de indices rechts worden uitgelijnd. Finaal moet het script ook *i*) het aantal anagrammen, *ii*) het totaal aantal woorden die een anagram hebben, *iii*) de langste woorden die anagrammen hebben en *iv*) de verzameling anagrammen die het meeste woorden bevat uitschrijven naar standaard uitvoer. Zorg ervoor dat het **awk**-script een functie `hash` bevat die voor een gegeven tekenreeks de hash-waarde berekent zoals hierboven werd beschreven. Als we het script bijvoorbeeld uitvoeren op het voorbeeldbestand `dictionary.txt`, dan moet de volgende uitvoer gegenereerd worden.

```
1: Ontario,oration
2: ailments,aliments
...
2521: callers,cellars,recalls
...
2532: retrievers,terriers
2533: porters,reports
2534: nursed,sunder

#anagrammen: 2534
#woorden: 5691
langste woord: algorithmically,logarithmically
meeste woorden: pares,parse,pears,rapes,reaps,spare,spear
```

Merk op dat de volgorde van de anagrammen niet noodzakelijk dezelfde hoeft te zijn als in bovenstaande uitvoer. Geef in je antwoordbestand ook de (ingekorte) uitvoer die gegenereerd wordt wanneer je je script uitvoert op het voorbeeldbestand `woordenboek.txt`.

### Opgave 3

Schrijf een macro `InitialiseerWortel` die een nieuw leeg werkblad aanmaakt in Excel, en die de eerste regel van dit werkblad opmaakt zoals aangegeven op onderstaande figuur.

	A	B	C	D	E	F	G
1		=		^2	+		WAAR

Hierbij worden enkel de cellen  $B1$ ,  $D1$ ,  $E1$  en  $G1$  aangepast. In de eerste drie cellen komt telkens de aangegeven tekst, die gecentreerd in de cel geplaatst wordt. Cel  $G1$  bevat een logische formule die aangeeft of de vergelijking  $A1 = C1 * C1 + F1$  daadwerkelijk opgaat. Op deze cel is ook een voorwaardelijke opmaak van toepassing. Vult een gebruiker na het toepassen van deze macro  $A1$ ,  $C1$  en  $F1$  op met getallen, dan kleurt de achtergrond van cel  $G1$  automatisch groen of rood naargelang de ingevulde bewerking klopt of niet.

	A	B	C	D	E	F	G
1	134	=	11	^2	+	9	ONWAAR

Nu bestaat er een methode om de vierkantswortel van een getal met de hand uit te rekenen. Deze methode vertoont sterke gelijkenissen met de staartdeling. Om de methode te illustreren zullen we de vierkantswortel berekenen van het getal 226567.

Initieel splitsen we het getal waarvan de vierkantswortel moet berekend worden op in cijferparen, waarbij we aan de rechterkant beginnen. Voor getallen met een oneven aantal cijfers blijft er links slechts één enkel cijfer over, wat geen enkele invloed heeft op de verdere berekeningen. Het voorbeeldgetal wordt dus op de volgende manier opgesplitst in cijferparen:  $22 \cdot 65 \cdot 67$ . Elk paar cijfers zal overeenkomen met een cijfer in de vierkantswortel van het opgegeven getal.

De berekening van de vierkantswortel start bij het meest linkse paar, net zoals bij een gewone staartdeling. In ons voorbeeld is dit het paar 22. We bepalen vervolgens het grootste kwadraat dat kleiner is of gelijk aan het getal gevormd door dit paar. In dit geval is het grootste kwadraat dus  $4^2 = 16$ . Boven het eerste paar schrijven we de vierkantswortel van het kwadraat (4), wat meteen ook het eerste cijfer van de vierkantwortel van het opgegeven getal aangeeft. Het kwadraat zelf (16) schrijven we onder het eerste paar en we trekken de twee waarden van elkaar af ( $22 - 16$ ), wat resulteert in de waarde 6.

Voor de volgende paren voeren we telkens opnieuw dezelfde procedure uit. We nemen het resultaat van de aftrekking, en voegen rechts de cijfers van het volgende paar toe. We noemen het zo bekomen getal  $D$ , wat in ons voorbeeld resulteert in het getal 665. Vermenigvuldig nu het getal dat bestaat uit alle cijfers van de voorlopig berekende vierkantwortel (het getal dat bestaat uit de cijfers dat we boven de opgesplitste paren schrijven) met twee, en noem dit getal  $Y$ . In ons voorbeeld is  $Y = 4 \times 2 = 8$ . Aan dit getal moet rechts nog het grootst mogelijke cijfer  $X$  ( $0 \leq X \leq 9$ ) worden toegevoegd waarvoor  $X \times YX \leq D$  (hierbij stelt  $YX$  de concatenatie van het getal  $Y$  en het cijfer  $X$  voor). In ons voorbeeld is  $X = 7$ , want  $7 \times 87 \leq 665 < 8 \times 88$ . We schrijven het cijfer  $X$  boven het huidige paar (65). De waarde van  $X \times YX$  (609) schrijven we onder de waarde van  $D$  en we trekken beide getallen van elkaar af. Dit resulteert voor ons voorbeeld in de waarde 56.

Er wordt gevraagd om ook nog een tweede macro `WortelTrekking` te schrijven die het getal uit de cel  $A1$  uitleest, en de worteltrekking  $\sqrt{A1}$  construeert zoals hierboven staat beschreven. Deze macro moet eveneens de opmaak verzorgen zoals in onderstaande figuur staat aangegeven. Deze geeft de oplossing weer van de worteltrekking van het gegeven getal 226567.

	A	B	C	D	E	F	G
1	226576	=	476	^2	+	0	WAAR
2							
3		4	7	6			
4		22	65	76			
5	4	-16					
6		6					
7			665				
8	87		-609				
9			56				
10				5676			
11	946			-5676			
12				0			

De vierkantswortel en de rest na worteltrekking worden door de macro `WortelTrekking` gekopieerd naar de corresponderende cellen op de eerste regel van het werkblad. Dankzij het voorbereidende werk van de macro `InitialiseerWortel` wordt dan ineens weergegeven of de worteltrekking correct werd uitgevoerd.

Ter informatie tonen we hieronder ook nog de oplossing van de berekening van de vierkantswortel van het getal 249418866.

	A	B	C	D	E	F	G
1	249418866	=	15793	^2	+	17	WAAR
2							
3		1	5	7	9	3	
4		2	49	41	88	66	
5	1	-1					
6		1					
7			149				
8	25		-125				
9			24				
10				2441			
11	307			-2149			
12				292			
13					29288		
14	3149				-28341		
15					947		
16						94766	
17	31583					-94749	
18						17	

## Opgave 4

Gegeven is een mailboxbestand `mailbox.mbox` in `mbox`-formaat. Het bestandsformaat `mbox` wordt algemeen gebruikt voor het opslaan van verzamelingen elektronische boodschappen. Alle boodschappen in een `mbox` mailbox worden in platte tekst achter elkaar opgeslagen in één enkel bestand. Het begin van elke boodschap wordt aangegeven door een regel waarvan de eerste vijf karakters bestaan uit het woord `From` gevolgd door een spatie, waarachter het emailadres komt te staan waarnaar niet afgeleverde boodschappen moeten teruggestuurd worden (het zogenaamde *bounce address*). Elke boodschap bestaat verder uit een `header` en een `body`, die van elkaar worden gescheiden door een lege regel. Let wel, de `body` zelf kan ook verschillende lege regels bevatten. Voor een technische beschrijving van het `mbox` formaat verwijzen we naar <http://qmail.org/qmail-manual-html/man5/mbox.html>.

In tegenstelling tot de Internetprotocollen die gebruikt worden voor het uitwisselen van email, werd het formaat voor het opslaan van email nooit formeel gedefinieerd via het RFC standaardiseringsmechanisme en bleef het onder de verantwoordelijkheid van de ontwerpers van email clients. Voor onderstaande opgaven mag je ervan uitgaan dat alle benodigde headers aanwezig zijn in elk van de emailboodschappen.

Gevraagd wordt:

1. Schrijf een `sed`-script `list.sed` dat een overzichtlijst genereert van alle boodschappen in een `mbox`-bestand. Elke regel in het overzicht geeft datum van verzending (`Date`-header), afzender (`From`-header) en onderwerp (`Subject`-header) van één mail, in die volgorde. Merk op dat het `mbox`-formaat zelf geen volgorde vastlegt voor de headerinformatie.

Verwijder uit de datum van verzending eventuele informatie over de tijdzone (alles na het plus-teken), en verwijder uit de afzender de dubbele aanhalingstekens en informatie die tussen kleiner dan en groter dan symbolen (`<...>`) staat. Beperk de lengte van de informatie over het onderwerp tot maximaal 60 karakters. Scheid de verschillende velden met een verticale streep (`'|'`).

De eerste drie regels die door het commando `gsed -f list.sed mailbox.mbox` gegenereerd worden, moeten er dan als volgt uitzien

```
Mon, 11 Feb 2018 10:51:07 | Yves Leterme | oahu!  
Fri, 13 May 1748 10:54:40 | Yves Leterme | Samson & Gert  
Tue, 06 Jul 2002 10:19:12 | Olivio Newton John | ch34p m3dz
```

2. Schrijf een `awk`-script `prettymail.awk` dat één enkele mail uit een `mbox`-bestand uitschrijft. De index van de desbetreffende mail wordt aan de hand van de parameter `nr` aan het script doorgegeven. Indien er een ongeldige waarde voor deze parameter wordt opgegeven, moet het script een gepaste foutboodschap genereren. Vóór de `body` van de mail moeten de drie regels van de header worden uitgeschreven die informatie bevatten over afzender, onderwerp en tijdstip van verzending (in die volgorde), gevolgd door een lege regel. Zo genereert het commando `gawk -v nr=7 -f prettymail.awk mailbox.mbox` de zevende mail uit de mailbox op de standaard uitvoer volgens onderstaand formaat.

```
From: Yves Leterme <guest@genix.ugent.be>  
Subject: oahu!  
Date: Mon, 11 Feb 2018 10:51:07 +0200 (CEST)
```

De oahu is een vogel

3. Schrijf een `awk`-script `spamfilter.awk` dat uit een `mbox`-bestand alle spammails verwijdert en enkel hammails (niet-spam) ongewijzigd uitschrijft naar standaard uitvoer. De regels die onderscheid maken tussen spam en ham zijn:

- het bestand `keywords_subject` bevat een lijst met reguliere expressies die niet mogen voorkomen in het onderwerp van de mail
- het bestand `keywords_body` bevat een lijst met reguliere expressies die niet mogen voorkomen in de body van de mail, maar wel in de header
- mails die door de mailserver werden gemarkeerd als spam, worden ook als dusdanig beschouwd (bekijk hiervoor de header-regel `X-Spam-Status`)

Zorg ervoor dat de spamfilter ook correct werkt als één of beide bestanden met trefwoorden leeg zijn of niet bestaan. Het commando `gawk -f spamfilter.awk mailbox.mbox` moet dus alle hammails ongewijzigd uitschrijven in het oorspronkelijk `mbox`-formaat.

4. Schrijf een `bash` shell script `mailtrooper.sh` dat bovenstaande scripts combineert tot een eenvoudige mail client. Aan dit script wordt een `mbox`-bestand als argument meegegeven. Na opstarten vraagt het script na de prompt `>>` naar gebruikerscommando's, die als volgt geïnterpreteerd worden:

- `q` of `Q` sluit het programma af
- `l` of `L` geeft een overzichtslijst van mails in het `mbox`-bestand. Gebruik hiervoor natuurlijk het script `list.sed`. Zorg ervoor dat mails in deze lijst oplopend genummerd worden, zodat gebruikers makkelijk de index van een bepaalde mail kunnen bepalen. Het is echter niet toegelaten om hiervoor het script `list.sed` aan te passen.
- alle andere invoer wordt beschouwd als parameter `nr` voor het script `prettymail.awk`

Zorg ervoor dat het script `mailtrooper.sh` bij het uitvoeren van deze commando's niet de volledige mailbox gebruikt, maar een spamvrije versie die verkregen wordt met behulp van het script `spamfilter.awk`. Het is echter niet toegelaten om een extra bestand aan te maken voor deze spamvrije mailbox. Het script `mailtrooper.sh` moet ook nagaan of alle databestanden en scriptbestanden die het gebruikt bestaan en leesbaar zijn, en of er daadwerkelijk een argument aan het script wordt doorgegeven.

---

---

## EXAMEN: Scriptingtalen

---

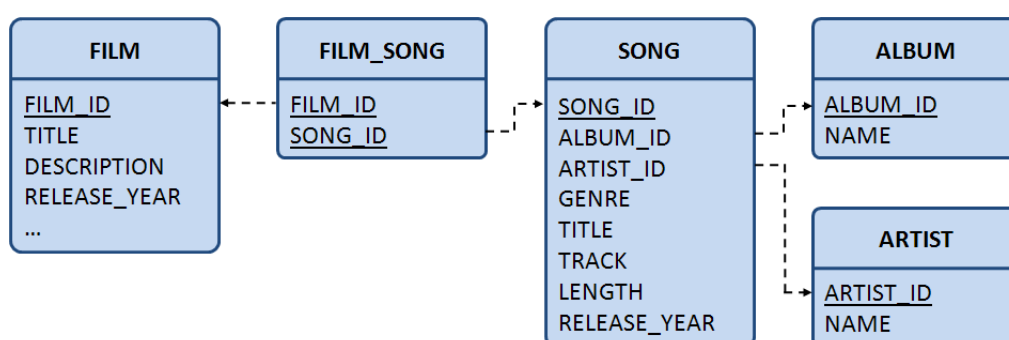
---

Prof. Dr. Peter Dawyndt  
1<sup>e</sup> Bachelor Informatica  
groep 3-4

maandag 23-06-2008, 14:00h  
academiejaar 2007-2008  
eerste zittijd

### Opgave 1

De DVD-verhuurketen heeft zijn Sakila databank uitgebreid, waardoor informatie kan worden opgezocht over soundtracks van films. Hiervoor werden vier extra tabellen aan de oorspronkelijke databank toegevoegd. Onderstaande figuur geeft een overzicht van de nieuwe tabellen, waarbij (samengestelde) primaire sleutels onderlijnd zijn en vreemde sleutels worden aangegeven met onderbroken pijlen.



De tabel **SONG** houdt een lijst bij van liedjes, waarbij aan elk liedje een uniek rangnummer toegekend wordt (kolom **SONG\_ID**). De kolom **ALBUM\_ID** is een vreemde sleutel die verwijst naar de gelijknamige kolom van de tabel **ALBUM**, en die aangeeft op welk album het liedje voorkomt. De kolom **ARTIST\_ID** is een vreemde sleutel die verwijst naar de gelijknamige kolom van de tabel **ARTIST**, en die aangeeft welke artiest het nummer geschreven heeft. De tabel **ALBUM** (resp. **ARTIST**) bevat verder enkel de naam van het album (resp. de artiest). Voorts geeft de tabel **SONG** ook nog informatie weer over de titel van het nummer (**TITLE**), het genre (**GENRE**), de plaats van het nummer op het album (**TRACK**), de lengte van het nummer (**LENGTH**) en het jaar waarin het werd uitgegeven (**RELEASE\_YEAR**). De tabel **FILM\_SONG** geeft aan welk nummer in welke film gebruikt wordt, en bevat daarvoor twee vreemde sleutels die verwijzen naar de primaire sleutels van de tabellen **FILM** en **SONG**.

De uitgebreide Sakila databank werd ingeladen in een MySQL RDBMS en is bevragebaar via de phpMyAdmin webapplicatie ([http://users.ugent.be/~dhollevo/scrita\\_sql/](http://users.ugent.be/~dhollevo/scrita_sql/)). Inloggen doe je in deze databank met je UGent account. Gevraagd wordt om SQL zoekopdrachten te formuleren die een antwoord bieden op onderstaande vragen. Zorg er telkens voor dat de kolommen van de resulterende tabel een zinvolle naam krijgen. Geef in je antwoordbestand ook aan hoeveel records de resultatentabel bevat.

1. Bereken het minimum, maximum en gemiddeld aantal liedjes per album.
2. Wat is het meest populair genre van alle liedjes die in de databank voorkomen. Geef ook aan hoeveel liedjes er tot dit genre behoren.
3. Geef een alfabetisch gerangschikte lijst van de titels van alle films waarin muziek gebruikt wordt van de artiest "The Prodigy". Let erop dat wegens spellingsinconsistenties, de naam van deze artiest in de volgende varianten voorkomt in de databank: "The Prodigy", "The prodigy" en "Prodigy".

4. Geef de namen van elk paar albums dat minstens twee liedjes bevat met dezelfde titel. Sorteert elk paar eerst op de eerste naam en dan op de tweede naam.

## Opgave 2

Schrijf een macro `InitialiseerDeling` die een nieuw leeg werkblad aanmaakt in Excel, en die de eerste regel van dit werkblad opmaakt zoals aangegeven op onderstaande figuur.

	A	B	C	D	E	F	G	H
1		=		*		+		TRUE
2								
3								

Hierbij worden enkel de cellen  $B1$ ,  $D1$ ,  $F1$  en  $H1$  aangepast. In de eerste drie cellen komt telkens de aangegeven tekst, die gecentreerd in de cel geplaatst wordt. Cel  $H1$  bevat een logische formule die aangeeft of de vergelijking  $A1 = C1 * E1 + G1$  daadwerkelijk opgaat. Op deze cel is ook een voorwaardelijke opmaak van toepassing. Vult een gebruiker na het toepassen van deze macro  $A1$ ,  $C1$ ,  $E1$  en  $G1$  op met getallen, dan kleurt de achtergrond van cel  $H1$  automatisch groen of rood naargelang de ingevulde bewerking klopt of niet.

	A	B	C	D	E	F	G	H
1	13	=	4	*	8	+	11	FALSE
2								
3								

Schrijf een tweede macro `StaatDeling` die de getallen uit de cellen  $A1$  en  $C1$  uitleest, en de staartdeling  $A1/C1$  construeert compleet met opmaak zoals in onderstaand voorbeeld staat aangegeven.

	A	B	C	D	E	F	G	H
1	123569	=	16	*	7723	+	1	TRUE
2								
3	1	2	3	5	6	9	16	
4	0						7723	
5	1							
6		12						
7		0						
8		12						
9			123					
10			-112					
11			11					
12				115				
13				-112				
14				3				
15					36			
16					-32			
17					4			
18						49		
19						-48		
20						1		
21								

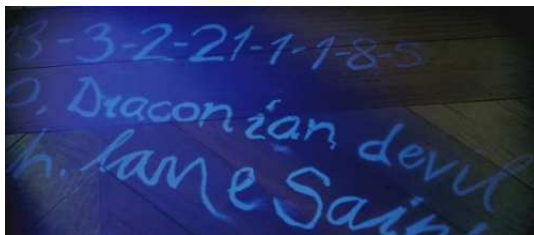


Het quotiënt en de rest van de deling worden door de macro `StaartDeling` gekopieerd naar de corresponderende cellen op de eerste regel van het werkblad. Dankzij het voorbereidende werk van de macro `InitialiseerDeling` wordt dan ineens weergegeven of de deling correct werd uitgevoerd. Zorg er ook voor dat de resultaten kloppen voor deeltal gelijk aan nul en negatief deeltal en/of deler.

**Hint:** Het aantal gebruikte kolommen in een staartdeling is gelijk aan het aantal cijfers in het deeltal plus één (voor de deler). De eerste kolom moet altijd de kolom *A* zijn. Vergeet niet de vorige staartdeling te wissen alvorens een nieuwe te construeren.

### Opgave 3

Een anagram is een woord dat of een zin die bestaat uit alle letters van een ander woord of een andere zin. Anagrammen worden vaak gebruikt als pseudoniem (*Leonardo da Vinci: o draconian devil; The Mona Lisa: oh lame saint*). Een eenvoudige manier om te bepalen of woorden anagrammen zijn, bestaat erin om een hash-waarde te berekenen zodat enkel anagrammen identieke waarden hebben. Voor de berekening van een hash-waarde kan je bijvoorbeeld aan elke letter van het alfabet een priemgetal toekennen ( $a=2, b=3, \dots, z=101$ ), en de waarden die corresponderen met elke letter van het originele woord met elkaar vermenigvuldigen. Commutativiteit van de vermenigvuldiging en uniciteit van de ontbinding in priemfactoren garanderen dat anagrammen een unieke hash-waarde hebben. Bij de berekening van de hash-waarde moeten enkel de letters uit het alfabet in rekening gebracht worden (geen leestekens, cijfers, etc.), en moeten deze niet-hoofdlettergevoelig behandeld worden. Zo resulteren de woorden `callers`, `cellars` en `RECALLS` allemaal in de hash-waarde 615461330, waardoor het dus anagrammen zijn.



Gevraagd wordt om een `awk`-script `anagram.awk` te schrijven dat alle combinaties van woorden die anagrammen zijn opzoekt in een gegeven woordenlijst. Het script leest de woordenlijst uit een bestand dat één enkel woord per regel bevat. Elke regel die door het script naar standaard uitvoer wordt weggeschreven moet alle woorden uit de woordenlijst bevatten die anagrammen van elkaar zijn. De anagrammen worden opeend genummerd, waarbij voor elke index vier posities gereserveerd worden en de indices rechts worden uitgelijnd. Finaal moet het script ook *i*) het aantal anagrammen, *ii*) het totaal aantal woorden die een anagram hebben, *iii*) de langste woorden die anagrammen hebben en *iv*) de verzameling anagrammen die het meeste woorden bevat uitschrijven naar standaard uitvoer. Zorg ervoor dat het `awk`-script een functie `hash` bevat die voor een gegeven tekenreeks de hash-waarde berekent zoals hierboven werd beschreven. Voor de berekening van de hash-waarden moet het script eerst automatisch een lijst van de eerste 26 priemgetallen opstellen. Je kan hiervoor gebruik maken van een zelfgedefinieerde functie `ispriem` die nagaat of een gegeven getal  $n$  een priemgetal is door te controleren dat het niet deelbaar is door  $2, 3, \dots, \sqrt{n}$ . Als we het script bijvoorbeeld uitvoeren op het voorbeeldbestand `dictionary.txt`, dan moet de volgende uitvoer gegenereerd worden.

```
1: Kyoto,Tokyo
2: budge,debug
...
14: danger,gander,garden,ranged
...
```

```
2532: aims,Siam
2533: credits,directs
2534: mundane,unnamed

#anagrammen: 2534
#woorden: 5691
langste woord: algorithmically,logarithmically
meeste woorden: pares,parse,pears,rapes,reaps,spare,spear
```

Merk op dat de volgorde van de anagrammen niet noodzakelijk dezelfde hoeft te zijn als in bovenstaande uitvoer. Geef in je antwoordbestand ook de (ingekorte) uitvoer die gegenereerd wordt wanneer je je script uitvoert op het voorbeeldbestand `woordenboek.txt`.

**Hint:** Hou rekening met het feit dat `awk` bij automatische conversie van getallen naar tekenreeksen voor grote getallen steeds de wetenschappelijke notatie zal gebruiken. Zo wordt de hash-waarde van het woord `abbreviation` omgezet naar de tekenreeks `1.44855e+14` en niet naar `144854651795436`. Je zal dus zelf moeten zorgen voor de gepaste conversie naar de laatste voorstelling.

## Opgave 4

In de bioinformatica is BLAST (*Basic Local Alignment Search Tool*) een veelgebruikt algoritme voor het vergelijken van nucleotide- of eiwitsequenties. Met een BLAST zoekopdracht kunnen onderzoekers een onbekende sequentie (de zogenaamde *query*sequentie) zeer snel vergelijken met een databank van sequenties, en op die manier sequenties opzoeken die sterke gelijkenissen vertonen met de querysequentie. Wanneer een onderzoeker bijvoorbeeld een onbekend gen ontdekt in de muis, dan zal hij typisch een BLAST zoekopdracht uitvoeren tegenover het menselijk genoom om te kijken of de mens gelijkaardige genen heeft.

Het bestand `blast.out` bevat het resultaat van een BLAST zoekopdracht. Dit bestand bevat alle sequenties (de zogenaamde *subject*sequenties) uit de databank die een similariteitsscore vertonen met de querysequentie die boven een bepaalde drempelwaarde ligt. De basisinformatie van elke subjectsequentie begint met een regel die als eerste karakter een groter dan symbool (`>`) heeft. Deze informatie kan eventueel over meerdere regels lopen, en wordt gevolgd door een regel die de lengte van de subjectsequentie vermeldt en een lege regel. Hierna volgt een lijst van subsequenties die zeer sterk op elkaar gelijken in de querysequentie en de subjectsequentie, volgens dalende similariteitsscore. Voor elke subsequentie worden eerst twee regels met statistieken weergegeven: de eerste regel bevat oa. de similariteitsscore (**Score**) en de significantie van de vergelijking (**Expect**), en de tweede regel bevat oa. het aantal identieke posities (**Identities**) en het aantal positieve posities (**Positives**). Daarna volgt een gedetailleerd alignement tussen de querysequentie en de subjectsequentie, per drie regels gesplitst over meerdere regels met daartussen telkens een lege regel.

Gevraagd wordt:

1. Schrijf een `sed`-script `blast_list.sed` dat een overzicht van de subjectsequenties genereert voor een gegeven BLAST output. Elke regel in het overzicht bestaat uit zes informatievelden die elk een vaste breedte hebben, en waarbij de verschillende velden door één enkele spatie van elkaar worden gescheiden. De eerste twee velden bevatten de informatie uit de regel die start met het `>` symbool in de originele BLAST output, waarbij de eerste spatie beschouwd wordt als scheidingsteken tussen de twee velden. Beide velden worden in het overzicht links uitgelijnd. Het eerste veld is 20 karakters breed en overtollige informatie wordt gewoon afgekapt. Het tweede veld is 40 karakters breed. Indien de originele informatie voor het tweede veld langer is dan 37 karakters dan wordt het op die positie afgekapt en wordt de rest vervangen door drie puntjes.

De overige informatievelen bevatten de volgende statistieken van de subsequentie met de hoogste similariteitsscore, waarbij de veldbreedte telkens tussen ronde haakjes staat: similariteitsscore (4), significantie van de vergelijking (8), procentueel aantal identieke posities (4) en procentueel aantal positieve posities (4). Deze velden worden allemaal rechts uitgelijnd.

De eerste vijf regels die door het commando `gsed -f blast_list.sed blast.out` gegenereerd worden, moeten er dan als volgt uitzien:

```
gnl|PID|e288614      (Z83122) R11A5.h [Caenorhabditis eleg... 173 1.3e-14 36% 79%
gi|1763275          (U73478) acidic nuclear phosphoprotei... 134 8.4e-11 33% 59%
gi|1769451          (X69465) ryanodine receptor 1 [Sus sc... 130 1.2e-09 46% 78%
gi|1752736          (D83006) gene required for phosphoyla... 121 8.5e-09 27% 66%
sp|P09741|TRT2_RABIT TROPONIN T, CARDIAC MUSCLE ISOFORMS      102 3.8e-08 29% 57%
```

- Schrijf een `awk`-script `blast_extract.awk` dat informatie over één enkele subjectsequentie uit een BLAST output extraheert. De index van de subjectsequentie wordt aan de hand van de parameter `hit` aan het script doorgegeven. Indien er een ongeldige waarde voor deze parameter wordt opgegeven, moet het script een gepaste foutboodschap genereren. De informatie over de gevraagde subjectsequentie moet ongewijzigd naar standaard uitvoer worden geschreven, en wordt daarbij beperkt tot informatie over de subquery met de hoogste similariteitsscore. Zorg er ook voor dat de laatste regel die wordt uitgeschreven geen lege regel is. Zo genereert het commando `gawk -v hit=4 -f blast_extract.awk blast.out` de vierde subjectsequentie uit de BLAST output op de standaard uitvoer, wat het volgende resultaat geeft.

```
gi|1752736 (D83006) gene required for phosphoylation of oligosaccharides/ has
      high homology with YJR061w [Saccharomyces cerevisiae]
      Length = 1178

Score = 121 (55.4 bits), Expect = 8.5e-09, P = 8.5e-09
Identities = 23/84 (27%), Positives = 56/84 (66%)

Query:   139 DEEEEEEEEEEEEEEEEEEDDDDDDEDGAEIQDDDEEGFDDEEEFDDEHDDDDLD 198
      +EEE++++EEEE +++EEEE+  +++++   E +   +E + +++ + ++ ++D +
Sbjct:  1062 EEEKKKKEEEEKKKKEEEEKKKKEEEEKKKQEEEEKKKKEEEEKKKQEEGKMKNEDEE 1121

Query:   199 NEENELEELEERVEARKKTTEKQS 222
      N++NE EE ++  E KK E+++
Sbjct:  1122 NKKNEDEEKKKNEEEEEKKKQEEKN 1145
```

- Schrijf een `bash` shell script `blast_viewer.sh` dat bovenstaande scripts combineert tot een eenvoudig programma om BLAST output te bekijken. Aan dit script wordt een BLAST outputbestand als argument meegegeven. Na opstarten vraagt het script na de prompt `>>` naar gebruikerscommando's, die als volgt geïnterpreteerd worden:

- `q` of `Q` sluit het programma af
- `l [n]` of `L [n]` genereert een overzicht van de eerste `n` subjectsequenties uit de BLAST output, met als standaardwaarde `n=5` indien deze parameter niet wordt opgegeven. Gebruik hiervoor natuurlijk het script `blast_list.sed`. Zorg ervoor dat de regels in het overzicht olopend genummerd worden, zodat gebruikers makkelijk de index van een bepaalde subjectsequentie kunnen bepalen. Het is echter niet toegelaten om hiervoor het script `blast_list.sed` aan te passen.
- alle andere invoer wordt beschouwd als parameter `hit` voor het script `blast_extract.awk`

Het script `blast_viewer.sh` moet nagaan of alle databestanden en scriptbestanden die het gebruikt bestaan en leesbaar zijn, en of er daadwerkelijk een argument aan het script wordt doorgegeven. Hieronder staat een voorbeeldsessie, waarbij het script `blast_viewer.sh` gebruikt wordt met de BLAST output `blast.out`.

```

bash$ blast_viewer.sh blast.out
>> L 10
  1. gnl|PID|e288614      (Z83122) R11A5.h [Caenorhabditis eleg... 173 1.3e-14 36% 79%
  2. gi|1763275          (U73478) acidic nuclear phosphoprotei... 134 8.4e-11 33% 59%
  3. gi|1769451          (X69465) ryanodine receptor 1 [Sus sc... 130 1.2e-09 46% 78%
  4. gi|1752736          (D83006) gene required for phosphoyla... 121 8.5e-09 27% 66%
  5. sp|P09741|TRT2_RABIT TROPONIN T, CARDIAC MUSCLE ISOFORMS 102 3.8e-08 29% 57%
  6. gi|1754694          (U81159) magnesium-dependent calcium ... 109 2.1e-06 36% 67%
  7. sp|Q08168|HRP_PLABE 58 KD PHOSPHOPROTEIN (HEAT SHOCK-RELA... 98 5.1e-06 66% 81%
  8. sp|Q10713|MPP1_HUMAN MITOCHONDRIAL PROCESSING PEPTIDASE AL... 94 4.2e-05 22% 44%
  9. gi|1762998          (U67260) RING-finger protein [Helicov... 92 7.4e-05 21% 47%
 10. sp|P02547|NFL_PIG   NEUROFILAMENT TRIPLET L PROTEIN (68 K... 93 0.00039 32% 57%
>> 1
gnl|PID|e288614 (Z83122) R11A5.h [Caenorhabditis elegans]
      Length = 906

Score = 173 (79.1 bits), Expect = 1.3e-14, Sum P(2) = 1.3e-14
Identities = 27/73 (36%), Positives = 58/73 (79%)

Query:  131 EGKTGNATDEEEEEEEEEEEEEEEEEEDDDDDDEDGAEIQDDDEEGFDDEEEFDDD 190
        E K  ++ +EE++++E+EEE E+EEEE+ED++D+++E++ +E +++D+E  + EEE D++
Sbjct:  576 ELKLLDSDEEEDDDDEDEEEDEDEEEDEDEEEDEEEENESEEEEDDEDEESEESDEE 635

Query:  191 EHDDDDLNEENE 203
        + ++++ D+ E E
Sbjct:  636 DEEEEEEEDDSEPE 648
>> q
bash$

```