
Computergebruik: EXAMEN

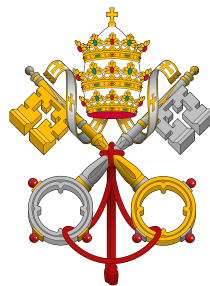
1^e Bachelor Informatica
prof. dr. Peter Dawyndt
groep 1

donderdag 10-01-2019, 08:30
academiejaar 2018-2019
eerste zittijd

Opgave 1 (pausen)

(10 pt)

De **paus** is het hoofd van de Rooms-Katholieke Kerk. De benaming paus is afkomstig van het Oudgriekse: *pappas* — later gelatiniseerd tot *papa*, in de betekenis van vader. De titel werd in de middeleeuwen ook opgevat als acroniem van het Latijnse *Petri apostoli potestatem accipiens* (de macht ontvangend van de apostel Petrus). De regeerperiode van een paus wordt **pontificaat** genoemd.



De paus draagt de titel Plaatsbekleder van Jezus Christus op Aarde, de opvolger van de heilige Petrus, de bisschop van Rome, het hoofd van de Rooms-Katholieke Kerk en is het staatshoofd van Vaticaanstad, het kleinste land ter wereld. In de uitoefening van zijn functie wordt de paus bijgestaan door de organen en instellingen van de Romeinse Curie. Buiten Rome wordt de paus vertegenwoordigd door nuntii bij regeringen en door apostolische delegaten in landen die geen diplomatieke betrekkingen onderhouden met de Heilige Stoel.

Het *Annuario Pontificio* (Italiaans voor Pauselijk Jaarboek) is het officiële jaarboek van de Heilige Stoel, die het sinds 1860 jaarlijks uitgeeft in het Italiaans. Het hoofdstuk *I Sommi Pontefici Romani* (de hoogste pausen van Rome) bevat een chronologische lijst van pausen, met uitzondering van de pausen die expliciet als tegenpausen aangeduid worden.

De eerste bisschop van Rome die zichzelf paus noemde, was Siricius (384–399), maar ook zijn voorgangers worden met terugwerkende kracht als paus aangeduid. Al sinds de 3^e eeuw kwam het geregeld voor dat er meerdere pausen tegelijk waren. In die gevallen erkent de Kerk één van de pausen als rechtmatig en reduceert de andere daarmee tot **tegenpaus**. In veel gevallen is echter niet eenduidig vast te stellen of een paus al dan niet rechtmatig was. Sommige pausen die in hun eigen tijd als rechtmatig golden, worden thans als tegenpaus beschouwd of om andere redenen niet meer meegerekend. Zo heeft het Vaticaan in 1961 “Stefanus (II)” uit de lijst geschrapt.

Pausen bleven de eerste eeuwen ook na hun wijding hun doopnaam gebruiken. De eerste paus die zijn naam veranderde is — voor zover bekend — Johannes II (533–535). Hij heette eigenlijk Mercurius, maar de verwijzing naar de heidense god Mercurius werd ongepast geacht. Uit soortgelijke motieven handelden Johannes XII (955–964), eigenlijk Octavianus, en Johannes XIV (983–984), eigenlijk Petrus. Vanaf Gregorius V (996–999) en Silvester II (999–1003), die beiden een Germaanse naam hadden, werd het gebruikelijk dat pausen bij hun wijding een nieuwe naam aannamen. De laatste paus die zijn doopnaam behield was Adrianus VI (1522–1523), die Adriaan heette. Voor Franciscus, die in 2013 paus werd, was Lando (913–914) de laatste paus die een naam koos die niet eerder was gedragen door minstens één paus. Min of meer een uitzondering betreft Johannes Paulus I (1978), die twee eerder gebruikte namen combineerde.

Opgave

Het tekstbestand `popes.txt` bevat een chronologische lijst van pausen, gebaseerd op de officiële lijst uit het *Annuario Pontificio*. Elke regel van het bestand bevat informatie over één paus, bestaande uit de volgende zeven velden die van elkaar worden gescheiden door een puntkomma (;): *i*) volgnummer van het pontificaat (**Pontiff number**), *ii*) startjaar van het pontificaat (**Pontificate start**), *iii*) eindjaar van het pontificaat (**Pontificate end**), *iv*) naam aangenomen bij wijding (**Pope name**), *v*) doopnaam (**Personal name**), *vi*) geboortedatum en geboorteplaats (**Birth data**) en *vii*) leeftijd bij begin en einde van het pontificaat (**Age at start/ end of papacy**). In het veld **Personal name** wordt de doopnaam gegeven in de taal van het geboorteland, eventueel gevolgd door een spatie en tussen ronde haakjes de Engelse versie van de naam als die verschilt van de oorspronkelijke naam. In het veld **Birth data** wordt de geboortedatum ofwel omschreven als *dag maand jaar* met *dag* en *jaar* in cijfers en *maand* in letters, of enkel als het geboortjaar (in cijfers). Dit wordt gevolgd door de geboorteplaats die nooit cijfers bevat. In het laatste veld (**Age at start/ end of papacy**) worden de twee leeftijden van elkaar gescheiden door een slash (/), eventueel voorafgegaan en gevolgd door één of meer spaties. De velden bevatten zelf nooit puntkomma's (;).

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 `popes.i.txt` (ZIP), die telkens de inhoud van het bestand bevatten nadat de *i*-de opdracht werd uitgevoerd.

1. Verwijder binnen elk veld alle fragmenten die tussen ronde haakjes staan, inclusief de ronde haakjes en alle spaties die vlak vóór het openende ronde haakje staan. Hierbij mag je ervan uitgaan dat er geen geneste ronde haakjes voorkomen (een fragment met ronde haakjes dat voorkomt binnen een fragment met ronde haakjes). Zo moet

```
1 | Pontiff number;Pontificate start;Pontificate end;Pope name;Personal name;Birth data;Age at start/ end of papacy
2 | 1;30;68;St Peter;šimon Kěpá (Simon Peter);1 Bethsaida, Galilea, Imperii Romaniae;29 / 63
3 | 2;64;79;St Linus;Linus (Lin);10 Volterrae, Italia, Roman Empire;54 / 66
4 | 3;76;91;St Anacletus;šAnákltos (Kltos) (Anaclet or Clet);25 Athenae, Achaea, Roman Empire;51 / 63
5 | ...
6 | 264;16 October 1978;2 April 2005;St John Paul II;Karol Józef ŹWojtya;20 May 1920 Wadowice, Poland;58 / 84
7 | 265;19 April 2005;28 Feb 2013;Benedict XVI;Joseph Aloisius Ratzinger;16 April 1927 MarktŹ am Inn, Bavaria, Germany;78 / 85
8 | 266;13 March 2013;present;Francis;Jorge Mario Bergoglio, S.J.;17 December 1936 Flores, Buenos Aires, Argentina;76 / -
```

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

```
1 | Pontiff number;Pontificate start;Pontificate end;Pope name;Personal name;Birth data;Age at start/ end of papacy
2 | 1;30;68;St Peter;šimon Kěpá;1 Bethsaida, Galilea, Imperii Romaniae;29 / 63
3 | 2;64;79;St Linus;Linus;10 Volterrae, Italia, Roman Empire;54 / 66
4 | 3;76;91;St Anacletus;šAnákltos;25 Athenae, Achaea, Roman Empire;51 / 63
5 | ...
6 | 264;16 October 1978;2 April 2005;St John Paul II;Karol Józef ŹWojtya;20 May 1920 Wadowice, Poland;58 / 84
7 | 265;19 April 2005;28 Feb 2013;Benedict XVI;Joseph Aloisius Ratzinger;16 April 1927 MarktŹ am Inn, Bavaria, Germany;78 / 85
8 | 266;13 March 2013;present;Francis;Jorge Mario Bergoglio, S.J.;17 December 1936 Flores, Buenos Aires, Argentina;76 / -
```

2. Splits het laatste veld (**Age at start/ end of papacy**) in twee velden met de leeftijd van de paus bij respectievelijk het begin en het einde van zijn pontificaat. Op de eerste regel moet ook de hoofding van dat veld gesplitst worden naar de benamingen **Age at start of papacy** en **Age at end of papacy**. We krijgen dan (`popes.2.txt`)

```
1 | Pontiff number;Pontificate start;Pontificate end;Pope name;Personal name;Birth data;Age at start of papacy;Age at end of
   | papacy
2 | 1;30;68;St Peter;šimon Kěpá;1 Bethsaida, Galilea, Imperii Romaniae;29;63
3 | 2;64;79;St Linus;Linus;10 Volterrae, Italia, Roman Empire;54;66
4 | 3;76;91;St Anacletus;šAnákltos;25 Athenae, Achaea, Roman Empire;51;63
5 | ...
6 | 264;16 October 1978;2 April 2005;St John Paul II;Karol Józef ŹWojtya;20 May 1920 Wadowice, Poland;58;84
7 | 265;19 April 2005;28 Feb 2013;Benedict XVI;Joseph Aloisius Ratzinger;16 April 1927 MarktŹ am Inn, Bavaria, Germany;78;85
8 | 266;13 March 2013;present;Francis;Jorge Mario Bergoglio, S.J.;17 December 1936 Flores, Buenos Aires, Argentina;76;
```

3. Splits het veld **Birth data** in twee velden met respectievelijk de geboortedatum en de geboorteplaats van de paus. Op de eerste regel moet ook de hoofding van dat veld gesplitst worden naar de benamingen **Birth year** en **Birthplace**. We krijgen dan (`popes.3.txt`)

```

1 | Pontiff number;Pontificate start;Pontificate end;Pope name;Personal name;Birth year;Birthplace;Age at start of papacy;Age at
  | end of papacy
2 | 1;30;68;St Peter;imon Kēpā;1;Bethsaida, Galilea, Imperii Romaniae;29;63
3 | 2;64;79;St Linus;Linus;10;Volterrae, Italia, Roman Empire;54;66
4 | 3;76;91;St Anacletus;Anákltos;25;Athenae, Achaea, Roman Empire;51;63
5 | ...
6 | 264;16 October 1978;2 April 2005;St John Paul II;Karol Józef Iwojtya;20 May 1920;Wadowice, Poland;58;84
7 | 265;19 April 2005;28 Feb 2013;Benedict XVI;Joseph Aloisius Ratzinger;16 April 1927;Marktl am Inn, Bavaria, Germany;78;85
8 | 266;13 March 2013;present;Francis;Jorge Mario Bergoglio, S.J.;17 December 1936;Flores, Buenos Aires, Argentina;76;

```

4. Behoud in het veld `Birth year` enkel het geboortjaar: waar er een geboortedag en geboortemaand vermeld worden, mogen die dus geschrapt worden. We krijgen dan (`popes.4.txt`)

```

1 | Pontiff number;Pontificate start;Pontificate end;Pope name;Personal name;Birth year;Birthplace;Age at start of papacy;Age at
  | end of papacy
2 | 1;30;68;St Peter;imon Kēpā;1;Bethsaida, Galilea, Imperii Romaniae;29;63
3 | 2;64;79;St Linus;Linus;10;Volterrae, Italia, Roman Empire;54;66
4 | 3;76;91;St Anacletus;Anákltos;25;Athenae, Achaea, Roman Empire;51;63
5 | ...
6 | 264;16 October 1978;2 April 2005;St John Paul II;Karol Józef Iwojtya;1920;Wadowice, Poland;58;84
7 | 265;19 April 2005;28 Feb 2013;Benedict XVI;Joseph Aloisius Ratzinger;1927;Marktl am Inn, Bavaria, Germany;78;85
8 | 266;13 March 2013;present;Francis;Jorge Mario Bergoglio, S.J.;1936;Flores, Buenos Aires, Argentina;76;

```

5. Sorteert de pausen eerst volgens stijgende leeftijd op het moment dat hun pontificaat startte en daarna volgens dalende duur van hun pontificaat. We krijgen dan (`popes.5.txt`)

```

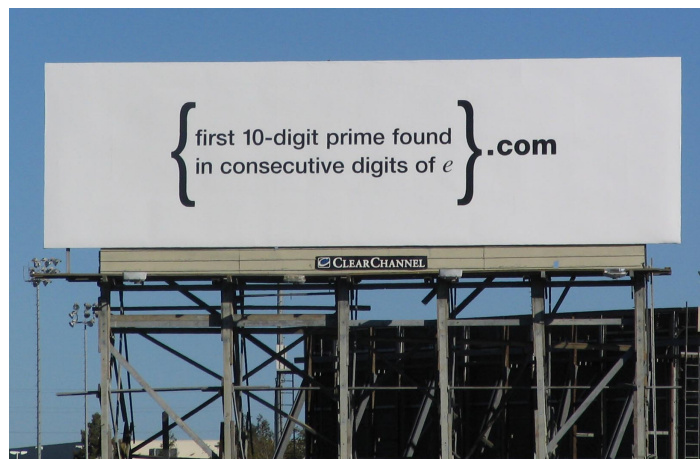
1 | Pontiff number;Pontificate start;Pontificate end;Pope name;Personal name;Birth year;Birthplace;Age at start of papacy;Age at
  | end of papacy
2 | 130;16 Dec 955;6 Dec 963;John XII;Ottaviano di Tuscolo;937;Rome, Papal States;18;26
3 | 145;21 October 1032;31 Dec 1044;Benedict IX;Teofilatto di Tuscolo;1012;Rome, Papal States;20;32
4 | 125;15 March 931;Dec 935;John XI;Giovanni di Tuscolo;910;Rome, Papal States;21;25
5 | ...
6 | 192;5 July 1294;13 Dec 1294;St Celestine V;Pietro Angelerio;1207;Sant'Angelo Limosano, Kingdom of Sicily;85;87
7 | 112;11 April 896;26 April 896;Boniface VI;Bonifacio;806;Rome, Papal States;90;90
8 | 79;27 June 678;10 Jan 681;St Agatho;Agáthn;577;Panormus, Sicilia, Byzantine Empire;101;104

```

Opgave 2 (auditie)

(10 pt)

In 2004 verscheen op verschillende plaatsen in de Verenigde Staten een mysterieus reclamebord:



De meeste mensen weten wel dat de wiskundige constante e het grondtal is van de natuurlijke logaritme, maar zoeken naar priemgetallen van 10 cijfers is geen eenvoudige opdracht. Gelukkig is het eerste priemgetal van 10 cijfers in e al te vinden vanaf positie 101:

```

2.718281828459045235360287471352662497757247093699959574966967627724076630353
547594571382178525166427427466391932003059921817413596629043572900334295260
595630738132328627943490763233829880753195251019011573834187930702154089149
934884167509244761460668082264800168477411853742345442437107539077744992069
551702761838606261331384583000752044933826560297606737113200709328709127443

```

Wie het mysterie had opgelost en de website <http://7427466391.com> bezocht, vond er een nog moeilijker probleem. De oplossing van dat probleem leidde naar een pagina bij Google Labs ... waarop een uitnodiging stond om een CV in te dienen.

Opgave

We zijn niet op zoek naar priemgetallen maar naar iets eenvoudiger patronen die voorkomen in de cijfers van de wiskundige constante e . Elke regel van het tekstbestand `digits.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 alle reeksen van opeenvolgende cijfers. De patronen die in het bestand staan zijn telkens de volgende 60 decimale cijfers van de wiskundige constante e . De verzameling \mathcal{W} bevat alle woorden die enkel bestaan uit kleine letters. Gevraagd wordt:

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

(a) $\mathcal{P}_1 = \{p \in \mathcal{P} \mid \text{som van eerste en laatste cijfer van } p \text{ is gelijk aan } 12\}$

voorbeeld: $\underline{3}12189912421733955068877864313075002482336183273872969737659 \in \mathcal{P}_1$
 $\underline{3}1745443290611383442776822551195335945096454297496538730055\underline{2} \notin \mathcal{P}_1$

(b) $\mathcal{P}_2 = \{p \in \mathcal{P} \mid p \text{ bevat quadruplet}^1 \text{ dat direct gevolgd wordt door omgekeerde quadruplet}\}$

¹ een **quadruplet** is een reeks van vier opeenvolgende cijfers; het **omgekeerde quadruplet** heeft de cijfers in omgekeerde volgorde

voorbeeld: $182546821879197195775913646990334225861823363434184764460488 \in \mathcal{P}_2$
 $817395598341290884664992191254332273554246324185540434071023 \notin \mathcal{P}_2$

(c) $\mathcal{P}_3 = \{p \in \mathcal{P} \mid \text{alle tripletten}^2 \text{ van } p \text{ zijn verschillend}\}$

² een **triplet** is een reeks van drie opeenvolgende cijfers; tripletten van p kunnen elkaar overlappen

voorbeeld: $915333276744766778516698071719343932220310951004535113777124 \in \mathcal{P}_3$
 $720384\underline{5}043224435400787467494566964553108820620025380667545\underline{00} \notin \mathcal{P}_3$

(d) $\mathcal{P}_4 = \{p \in \mathcal{P} \mid p \text{ bevat een oneven aantal even cijfers}\}$

voorbeeld: $\underline{6}17592982744463332097958931321177673952240461419211613357318 \in \mathcal{P}_4$
 $\underline{0}58982406762373406449857531714631332550639355128465018723271 \notin \mathcal{P}_4$

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 3 (size matters)

(10 pt)

In juni 2017 gaf de *California Olive Association* nogal angstaanjagende namen aan de groottes van olijven:

NAAM	AANTAL OLIJVEN PER POND
<i>Standard</i>	120–135
<i>Medium</i>	105–120
<i>Large</i>	90–105
<i>Extra Large</i>	75–90
<i>Mammoth</i>	65–75
<i>Colossal</i>	55–65
<i>Giant</i>	45–55

In de loop der jaren hebben ze ook *Jumbo*, *Supercolossal* en *Special Supercolossal* aan de lijst met namen toegevoegd.



Het duurde tot de jaren '70 vooraleer de regering stappen zette om verdere wildgroei een halt toe te roepen:

The Department of Agriculture feels that most people would not be able to figure out which are the larger olives, except at the range of smaller sizes, whose names are the more straightforward.

Opgave

JPEG is een bestandsindeling die gebruikt wordt voor het opslaan van rechthoekige rasterafbeeldingen in digitale vorm. De bestandsinformatie die wordt uitgeschreven door het Unix commando `file` bevat voor een JPEG-bestand onder andere de dimensie van de afbeelding (500x375 in onderstaand voorbeeld):

```
$ file 0.jpg
0.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16,
baseline, precision 8, 500x375, frames 3
```

Schrijf een `bash` shell script `jpegdim` waaraan de padnaam van een directory moet doorgegeven worden. Het shell script moet alle JPEG-bestanden onder de gegeven directory (dus inclusief alle onderliggende subdirectories) hernoemen. Elk JPEG-bestand `image.ext` (waarbij er geen punt voorkomt in de extensie `ext`) moet hernoemd worden naar `image.dim.ext` (waarbij `dim` de dimensie van de afbeelding is zoals die wordt weergegeven door het Unix commando `file`). Het shell script moet ervan uitgaan dat alle JPEG-bestanden een extensies `.jpg` of `.jpeg` hebben, waarbij geen onderscheid mag gemaakt worden tussen hoofdletters en kleine letters.

Voorbeeld

In onderstaande voorbeeldsessie gaan we ervan uit dat de directory `reception` (`reception.zip`) zich in de huidige directory bevindt.

```
$ tree reception/
reception/
  champagne
    0.jpg
    1.jpeg
    2.jpg
    3.jpeg
    4.JPG
    5.jpg
    6.jpg
    7.jpg
    8.JPeg
    9.jpeg
  olives
    0.jpg
    1.jpeg
    2.jpg
    3.jpeg
    4.JPG
    5.jpg
    6.jpg
    7.jpg
    8.JPeg
    collection.jpeg

2 directories, 20 files
$ jpegdim reception/
$ tree reception/
reception/
  champagne
    0.224x225.jpg
    1.225x225.jpeg
    2.252x200.jpg
    3.275x183.jpeg
    4.216x233.JPG
    5.275x183.jpg
    6.225x225.jpg
    7.247x204.jpg
    8.219x230.JPeg
    9.225x225.jpeg
  olives
    0.275x183.jpg
    1.225x225.jpeg
    2.183x275.jpg
    3.259x194.jpeg
    4.225x225.JPG
    5.259x194.jpg
    6.275x183.jpg
    7.275x183.jpg
    8.300x168.JPeg
    collection.275x183.jpeg

2 directories, 20 files
$
```

Alle directories die gebruikt worden voor het testen van de ingezonden oplossingen zijn gebundeld in `test.zip`.

Opgave 4 (Homer backs into things)

(20 pt)

Op 17 maart 1994 werd de aflevering “*Homer Loves Flanders*” van de Simpsons voor het eerst uitgezonden. Daarin komt een scène voor waarin het hoofdpersonage Homer Simpson aan zijn burens — de familie Flanders — vraagt of ze met hem op stap willen gaan. Als ze antwoorden dat ze daar geen zin in hebben, antwoordt Homer met “Oh, ok” voordat hij in de struiken verdwijnt.



De scène hierboven werd op 8 december 2010 vastgelegd in een animated GIF die is uitgegroeid tot één van de bekendste Internet memes. De GIF wordt gebruikt als uitdrukking van schaamte en werd vaak bewerkt tot Homer (of iemand anders) die in andere dingen verdwijnt.

Opgave

GIF (*Graphics Interchange Format*) is een bestandsformaat voor het opslaan van rasterafbeeldingen in digitale vorm. Het formaat ondersteunt kleuren, verschillende resoluties, animatie en een transparante achtergrond. Bij een **animated GIF** worden verschillende afbeeldingen achter elkaar opgeslagen in hetzelfde GIF-bestand. Deze afbeeldingen worden de **frames** van de animated GIF genoemd. Animatie ontstaat door deze frames snel na elkaar af te spelen. Door na het laatste frame terug het eerste frame weer te geven, kan een animated GIF zichzelf eindeloos herhalen.

ImageMagick is een open source bibliotheek die bestaat uit verschillende Unix commando's voor het weergeven, converteren en bewerken van digitale afbeeldingen (zowel rasterafbeeldingen als vectorafbeeldingen). De bibliotheek bevat onder andere het commando `convert` dat als volgt kan gebruikt worden om drie afbeeldingen `0.png`, `1.png` en `2.png` in PNG-formaat te bundelen tot één animated GIF `animated.gif`:

```
$ convert frame0.png frame1.png frame2.png animated.gif
```

Hetzelfde commando kan als volgt gebruikt worden om de frames van een animated GIF `animated.gif` op te slaan als individuele bestanden in PNG-formaat:

```
$ convert -coalesce animated.gif "frame%d.png"
$ ls
frame0.png frame1.png frame2.png
```

Hierbij vormt het laatste argument ("`frame%d.png`") een template die aangeeft hoe de bestanden met de individuele frames moeten genoemd worden. Hierbij worden de frames genummerd vanaf nul en wordt de plaatshouder `%d` ingevuld met het volgnummer van het frame in het GIF-bestand. Het commando `identify` uit de ImageMagick bibliotheek kan dan weer gebruikt worden om te bepalen hoeveel frames een animated GIF heeft.

Schrijf een `bash` shell script `animate` dat kan gebruikt worden om bewerkingen uit te voeren op animated GIF-bestanden. Aan het shell script moeten altijd minstens twee argumenten doorgegeven worden: *i*) de naam van een subcommando en *ii*) de padnaam van een (bestaand of nieuw aan te maken) GIF-bestand. Het shell script moet de volgende vijf subcommando's ondersteunen:

bundle Als extra argumenten moeten aan dit subcommando de padnamen van één of meer PNG-bestanden doorgegeven worden. Als er geen extra argumenten doorgegeven worden dan moet een boodschap met de signatuur van het subcommando (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 2`. Als één van de extra argumenten geen gewoon bestand is of niet leesbaar is, dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 3`. Anders moet het shell script de gegeven PNG-bestanden bundelen tot één animated GIF, die wordt opgeslagen op de bestandslocatie die als tweede argument werd doorgegeven aan het shell script.

unpack Als extra argument kan aan dit subcommando optioneel nog een padnaam van een directory doorgegeven worden. Als er meer dan één extra argument doorgegeven wordt dan moet een boodschap met de signatuur van het subcommando (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 2`. Anders moet het shell script de frames van het gegeven animated GIF-bestand opslaan als individuele PNG-bestanden in de gegeven directory. Hierbij moeten de opeenvolgende frames benoemd worden als `0.png`, `1.png`, `2.png`, Als er geen directory gegeven werd dan moeten de bestanden in de huidige directory opgeslagen worden. Als de gegeven directory nog niet bestond dan moet die door het shell script aangemaakt worden.

sample Als extra argumenten moeten de padnaam van een tweede GIF-bestand doorgegeven worden, gevolgd door één of meer volgnummers van frames in het eerste GIF-bestand (waarbij de opeenvolgende frames genummerd worden vanaf nul). Als er minder dan twee extra argumenten doorgegeven worden dan moet een boodschap met de signatuur van het subcommando (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 2`. Als één van de gegeven frames buiten het bereik van de volgnummers van het eerste GIF-bestand ligt dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 3`. Anders moet het shell script een nieuw GIF-bestand aanmaken op de gegeven bestandslocatie (padnaam van tweede GIF-bestand), waarvan de frames bestaan uit de geselecteerde frames van het gegeven GIF-bestand (padnaam van eerste GIF-bestand) in de volgorde waarop de volgnummers als argument doorgegeven worden.

invert Als extra argument moet de padnaam van een tweede GIF-bestand doorgegeven worden. Als dat niet het geval is dan moet een boodschap met de signatuur van het subcommando (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 2`. Anders moet het shell script een nieuw GIF-bestand aanmaken op de gegeven bestandslocatie (padnaam van tweede GIF-bestand), waarvan de frames bestaan uit de frames van het gegeven GIF-bestand (padnaam van eerste GIF-bestand), maar dan in omgekeerde volgorde.

mirror Als extra argument moet de padnaam van een tweede GIF-bestand doorgegeven worden. Als dat niet het geval is dan moet een boodschap met de signatuur van het subcommando (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 2`. Anders moet het shell script een nieuw GIF-bestand aanmaken op de gegeven bestandslocatie (padnaam van tweede GIF-bestand), waarvan de frames bestaan uit de frames van het gegeven GIF-bestand (padnaam van eerste GIF-bestand) gevolgd door de frames van het gegeven GIF-bestand maar dan in omgekeerde volgorde. Het resultaat is dat de frames van het gegeven GIF-bestand in het nieuwe GIF-bestand van voor naar achter en van achter naar voor worden afgespeeld.

Naast alle foutafhandeling die we reeds bij de individuele subcommando's vermeld hebben, moet het shell script ook de volgende foutafhandeling voorzien:

- als er minder dan twee argumenten worden doorgegeven aan het shell script, als het eerste argument bestaat uit de optie `-h` of als het eerste argument geen geldig subcommando is, dan moet een overzicht met de signatuur van het commando en zijn subcommando's (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 1`
- voor alle subcommando's behalve `bundle`: als de padnaam die als tweede argument wordt doorgegeven geen gewoon bestand is dat kan uitgelezen worden, dan moet een boodschap met de signatuur van het subcommando (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stdout` en moet het shell script eindigen met `exit status 2`

Als er zich geen fouten voordoen dan moet het shell script eindigen met `exit status 0`. Als het shell script tijdelijke bestanden of directories aanmaakt, dan moet het er ook voor zorgen dat die bestanden of directories terug verwijderd worden vóór het shell script eindigt.

Voorbeeld

Onderstaande voorbeeldsessie geeft aan hoe het shell script `animate` moet kunnen gebruikt worden. Hierbij gaan we ervan uit dat het tekstbestand `homer.gif` zich in de huidige directory bevindt.

```
$ animate -h
animate [-h] [<action> <GIF> ...]

ACTIONS
  animate bundle <GIF> <FRAME> ...
  animate unpack <GIF> [<DIR>]
  animate sample <GIF> <GIF> <FRAME> ...
  animate invert <GIF> <GIF>
  animate mirror <GIF> <GIF>
$ echo $?
1
$ animate unpack homer.gif homer/
$ ls homer/
0.png  15.png  21.png  28.png  34.png  40.png  47.png  53.png
1.png  16.png  22.png  29.png  35.png  41.png  48.png  54.png
10.png 17.png  23.png  3.png   36.png  42.png  49.png  55.png
11.png 18.png  24.png  30.png  37.png  43.png  5.png   6.png
12.png 19.png  25.png  31.png  38.png  44.png  50.png  7.png
13.png 2.png   26.png  32.png  39.png  45.png  51.png  8.png
14.png 20.png  27.png  33.png  4.png   46.png  52.png  9.png
$ animate bundle homer.bundled.gif homer/0.png homer/10.png homer/20.png
$ animate sample homer.gif homer.sampled.gif 0 10 20 30 40 50
$ animate invert homer.gif homer.inverted.gif
$ animate mirror homer.gif homer.mirrored.gif
$ ls
animate  homer.bundled.gif  homer.inverted.gif  homer.sampled.gif
homer    homer.gif          homer.mirrored.gif
```

Onderstaande voorbeeldsessie toont enkele voorbeelden van de foutafhandeling die het shell script `animate` moet voorzien.

```
$ animate bundle homer.gif
animate bundle <GIF> <FRAME> ...
$ echo $?
2
$ animate bundle homer.bundled.gif homer/0.png XXX homer/20.png
animate bundle: file XXX does not exist
$ echo $?
3
$ animate unpack homer.gif homer/ XXX
```

```
animate unpack <GIF> [<DIR>]
$ echo $?
2
$ animate sample homer.gif homer.sampled.gif
animate sample <GIF> <GIF> <FRAME> ...
$ echo $?
2
$ animate sample homer.gif homer.sampled.gif 0 10 1000 1000000
animate sample: frames must be in range [0, 55]
$ echo $?
3
$ animate invert homer.gif
animate invert <GIF> <GIF>
$ echo $?
2
$ animate mirror homer.gif
animate mirror <GIF> <GIF>
$ echo $?
2
$ animate mirror marge.gif marge.mirrored.gif
animate mirror <GIF> <GIF>
$ echo $?
2
```