
Computergebruik: EXAMEN

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

dinsdag 21-08-2018, 14:00
academiejaar 2017-2018
tweede zittijd

Opgave 1 (fastfood)

(10 pt)

Toen Walter A. Anderson en Edgar Waldo "Billy" Ingram in 1921 de White Castle fastfoodketen oprichtten in de Verenigde Staten, was het vermoedelijk de eerste maar zeker niet de laatste van zijn soort. Ondertussen is de fastfoodindustrie een zeer grote, lucratieve industrie geworden, die maar blijft groeien omdat mensen steeds minder tijd nemen om zelf te koken. Amerikaanse fastfoodketens zoals McDonald's (opgericht in 1940) en KFC (opgericht in 1952) zijn multinationals geworden met vestigingen over de hele wereld.



Opgave

Het tekstbestand `fastfood.txt` bevat de locatie van enkele Amerikaanse fastfoodrestaurants. Elke regel van het bestand bevat informatie over één fastfoodrestaurant, bestaande uit tien of meer velden die van elkaar gescheiden worden door tabs: *i*) straatnaam, *ii*) stad, *iii*) land, *iv*) unieke code, *v*) breedtegraad, *vi*) lengtegraad, *vii*) naam, *viii*) postcode, *ix*) provincie en *x*) URLs van websites. Als er meerdere URLs van websites gegeven worden, dan zijn die ook van elkaar gescheiden door tabs. De eerste regel van het tekstbestand bevat de namen van de velden, van elkaar gescheiden door tabs. Je mag ervan uitgaan dat de informatievelden 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 `fastfood.i.txt` (ZIP), die telkens de inhoud van het bestand bevatten nadat de *i*-de opdracht werd uitgevoerd.

1. Vervang alle tabs door puntkomma's, behalve tussen twee URLs van websites. Deze vervanging moet ook op de eerste regel doorgevoerd worden. Zo moet

```
1 | address city      country keys      latitude      longitude      name           postalCode     province      websites
2 | 324 Main St      Massena US        us/ny/massena/324mainst/-1161002137  44.9213 -74.89021     McDonald's     13662
3 | NY              http://mcdonalds.com http://www.mcdonalds.com/?cid=RF:YXT_FM:TP::Yext:Referral
3 | 530 Clinton Ave Washington Court House US        us/oh/washingtoncourthouse/530clintonave/-791445730 39.53255
   | -83.44526       Wendy's 43160 OH        http://www.wendys.com
```

```

4 | 408 Market Square Dr      Maysville      US      us/ky/maysville/408marketsquaredr/1051460804      38.62736
   | -83.79141      Frisch's Big Boy      41056      KY      http://www.frischs.com      https://www.frischs.com/location/
   | maysville-ky/
5 | 6098 State Highway 37      Massena      US      us/ny/massena/6098statehighway37/-1161002137      44.95008      -74.84553
   | McDonald's      13662      NY      http://mcdonalds.com      http://www.mcdonalds.com/?cid=RF:YXT_FM:TP::Yext:
   | Referral
6 | ...

```

bijvoorbeeld omgezet worden naar (fastfood.1.txt)

```

1 | address;city;country;keys;latitude;longitude;name;postalCode;province;websites
2 | 324 Main St;Massena;US;us/ny/massena/324mainst/-1161002137;44.9213;-74.89021;McDonald's;13662;NY;http://mcdonalds.com
   | http://www.mcdonalds.com/?cid=RF:YXT_FM:TP::Yext:Referral
3 | 530 Clinton Ave;Washington Court House;US;us/oh/washingtoncourthouse/530clintonave/-791445730;39.53255;-83.44526;Wendy's;
   | 43160;OH;http://www.wendys.com
4 | 408 Market Square Dr;Maysville;US;us/ky/maysville/408marketsquaredr/1051460804;38.62736;-83.79141;Frisch's Big Boy
   | ;41056;KY;http://www.frischs.com      https://www.frischs.com/location/maysville-ky/
5 | 6098 State Highway 37;Massena;US;us/ny/massena/6098statehighway37/-1161002137;44.95008;-74.84553;McDonald's;13662;NY;
   | http://mcdonalds.com      http://www.mcdonalds.com/?cid=RF:YXT_FM:TP::Yext:Referral
6 | ...

```

2. Verwijder de kolom met de unieke code (keys) en behoud enkel de eerste URL (website). Het verwijderen van de kolom moet ook op de eerste regel doorgevoerd worden. We krijgen dan (fastfood.2.txt)

```

1 | address;city;country;latitude;longitude;name;postalCode;province;websites
2 | 324 Main St;Massena;US;44.9213;-74.89021;McDonald's;13662;NY;http://mcdonalds.com
3 | 530 Clinton Ave;Washington Court House;US;39.53255;-83.44526;Wendy's;43160;OH;http://www.wendys.com
4 | 408 Market Square Dr;Maysville;US;38.62736;-83.79141;Frisch's Big Boy;41056;KY;http://www.frischs.com
5 | 6098 State Highway 37;Massena;US;44.95008;-74.84553;McDonald's;13662;NY;http://mcdonalds.com
6 | ...

```

3. Behoud enkel de domeinnaam uit de URL van de website. Zo moet

```

1 | ...
2 | 4182 Tonya Trl;Hamilton;US;39.4176;-84.47641;Domino's Pizza;45011;OH;https://www.dominos.com/en/?redirect=homepage
3 | ...
4 | 613 W Center St;Sheridan;US;34.30687;-92.40784;SONIC Drive In;72150;AR;https://locations.sonicdrivein.com/ar/sheridan
   | /613-w--center-street.html
5 | ...
6 | 1423 Southwest Pkwy;Wichita Falls;US;33.85371;-98.50065;McDonald's;76302;TX;http://www.mcdonalds.com/?cid=RF:YXT_FM:TP
   | ::Yext:Referral
7 | ...

```

bijvoorbeeld omgezet worden naar (fastfood.3.txt)

```

1 | ...
2 | 4182 Tonya Trl;Hamilton;US;39.4176;-84.47641;Domino's Pizza;45011;OH;www.dominos.com
3 | ...
4 | 613 W Center St;Sheridan;US;34.30687;-92.40784;SONIC Drive In;72150;AR;locations.sonicdrivein.com
5 | ...
6 | 1423 Southwest Pkwy;Wichita Falls;US;33.85371;-98.50065;McDonald's;76302;TX;www.mcdonalds.com
7 | ...

```

4. Verplaats de kolom met de naam van het restaurant (name) zodat het de eerste kolom wordt. Deze verplaatsing moet ook op de eerste regel doorgevoerd worden. We krijgen dan (fastfood.4.txt)

```

1 | name;address;city;country;latitude;longitude;postalCode;province;websites
2 | McDonald's;324 Main St;Massena;US;44.9213;-74.89021;13662;NY;mcdonalds.com
3 | Wendy's;530 Clinton Ave;Washington Court House;US;39.53255;-83.44526;43160;OH;www.wendys.com
4 | Frisch's Big Boy;408 Market Square Dr;Maysville;US;38.62736;-83.79141;41056;KY;www.frischs.com
5 | McDonald's;6098 State Highway 37;Massena;US;44.95008;-74.84553;13662;NY;mcdonalds.com
6 | ...

```

5. Sorteert de fastfoodrestaurants in oplopende volgorde per locatie: eerst per land (country), dan per provincie (province), dan per stad (city) en tenslotte per straatnaam (address). Het sorteren moet gebeuren in de traditionele sorteervolgorde volgens de individuele waarde van de bytes. We krijgen dan (fastfood.5.txt)

```

1 | name;address;city;country;latitude;longitude;postalCode;province;websites
2 | McDonald's;1166 N Muldoon Rd;Anchorage;US;61.203697;-149.74467;99504;AK;mcdonalds.com
3 | Regal Cinemas Tikahatnu 16 IMAX & RPX;1299 N Muldoon Rd.;Anchorage;US;61.22662692;-149.7408595;99504;AK;www.regmovies.
   | com
4 | Arby's;320 W 5th Ave;Anchorage;US;61.217082;-149.887645;99501;AK;
5 | Arby's;800 E Dimond Blvd 025;Anchorage;US;61.14323525;-149.86685425;99515;AK;www.arbys.com
6 | ...

```

Opgave 2 (UUID)

(10 pt)

Een *universally unique identifier* (UUID) is een getal van 128 bits dat door computersystemen gebruikt wordt om wereldwijd de identiteit van objecten te kunnen bepalen. Daarom wordt dit concept soms ook aangeduid met de term *globally unique identifier* (GUID).

In tegenstelling tot de meeste andere identificatieprotocollen is het genereren van UUIDs niet afhankelijk van een centraal register en vraagt het ook geen coördinatie tussen de partijen die ze genereren. Als UUIDs volgens de standaardmethoden gegenereerd worden dan kunnen ze in de praktijk als wereldwijd uniek beschouwd worden. Hoewel het niet onmogelijk is dat dezelfde UUID meerdere keren gegenereerd wordt, ligt de kans daarop dicht genoeg bij nul om verwaarloosbaar te zijn.

Daarom kan iedereen onafhankelijk van anderen UUIDs genereren en gebruiken als wereldwijde identiteit van objecten. Er wordt immers met quasi zekerheid gegarandeerd dat de identiteit van de objecten geen duplicaat is van de identiteit die door iemand anders aan een ander object wordt toegekend. Daardoor hebben objecten die door onafhankelijke partijen met UUIDs geïdentificeerd worden een verwaarloosbare kans om door elkaar gehaald te worden als ze later in één enkele databank gecombineerd worden of als ze over hetzelfde communicatiekanaal verzonden worden.

De structuur en de tekstuele voorstelling van UUIDs worden vastgelegd in de RFC 4122 standaard. In de canonieke tekstuele voorstelling worden de zestien octetten (16×8 bits = 128 bits) van een UUID weergegeven in vijf groepen die van elkaar gescheiden worden door koppeltekens:

550e8400-e29b-41d4-a716-446655440000

De groepen bestaan respectievelijk uit 8, 4, 4, 4 en 12 hexadecimale cijfers, voor een totaal van 36 tekens (32 alfanumerieke tekens en vier koppeltekens). Onder de hexadecimale cijfers maken we onderscheid tussen cijfers (0–9) en letters (a–f). RFC 4122 legt ook vast dat kleine letters moeten gebruikt worden voor de hexadecimale cijfers.

Opgave

Elke regel van het tekstbestand `uuid.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 canonieke tekstuele voorstelling (volgens RFC 4122) van alle mogelijke UUIDs. 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) $\mathcal{P}_1 = \{p \in \mathcal{P} \mid \text{in elke groep van } p \text{ zijn alle hexadecimale cijfers verschillend}\}$

voorbeeld: `a91e35fd-b0d4-b95a-fdb0-f0d7b839c645` $\in \mathcal{P}_1$
`0fa05af9-87e9-49d0-bfdd-463e9993f90f` $\notin \mathcal{P}_1$

(b) $\mathcal{P}_2 = \{p \in \mathcal{P} \mid \text{de tweede en de derde groep van } p \text{ hebben geen enkel hexadecimaal cijfer gemeenschappelijk}\}$

voorbeeld: `6482cf60-dbf8-412c-a63c-d2b50e92ed6d` $\in \mathcal{P}_2$
`7b17ccef-f9f2-4597-aa0d-5f186d206a36` $\notin \mathcal{P}_2$

(c) $\mathcal{P}_3 = \{p \in \mathcal{P} \mid \text{in elke groep van } p \text{ staan de letters in strikt alfabetische volgorde}\}$

voorbeeld: `2beb1c38-8158-4448-a280-28c46a515b87` $\in \mathcal{P}_3$
`64efc3d9-448a-4902-a6c6-61849ac6c7ef` $\notin \mathcal{P}_3$

(d) $\mathcal{P}_4 = \{p \in \mathcal{P} \mid \text{er is een hexadecimaal cijfer dat in elke groep van } p \text{ minstens één keer voorkomt}\}$

voorbeeld: `6482cf60-dbf8-412c-a63c-d2b50e92ed6d` $\in \mathcal{P}_4$
`cd53e91b-4bce-4e58-ad28-9416cf468d71` $\notin \mathcal{P}_4$

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 standaard uitvoer 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 standaard uitvoer. Hierbij is het niet toegelaten om het woord w_j letterlijk uit te schrijven (bv. `echo w_j`).

Opgave 3 (rijksregisternummer)

(10 pt)

Het Belgisch Rijksregister is een databank met gegevens over elke inwoner van België. De databank wordt beheerd door de Federale Overheidsdienst Binnenlandse Zaken, maar de gegevens worden aangeleverd door de dienst Bevolking van de gemeenten. Kort na de geboorte krijgt elke inwoner een uniek **rijksregisternummer** waarmee een aantal administratieve gegevens in het Rijksregister terug te vinden zijn (domicilie, burgerlijke stand, nationaliteit). Dit nummer staat vermeld op de identiteitskaart en kan persoonlijk opgevraagd worden aan het loket van de dienst Bevolking.



Het rijksregisternummer bestaat uit 11 cijfers:

1. Een eerste groep van zes cijfers correspondeert met de **geboortedatum** in de volgorde jaar, maand en dag. Van het geboortjaar worden enkel de laatste twee cijfers gebruikt. De maand en de dag worden weergegeven met twee cijfers door eventueel een voorlooptul toe te voegen.
2. Een tweede groep van drie cijfers dient om onderscheid te maken tussen inwoners die op dezelfde dag geboren zijn. Het is een **dagteller** van de geboortes, waarbij voor vrouwen gebruikgemaakt wordt van even reeksnummers (001997) en voor mannen van oneven reeksnummers (002998).
3. Een derde groep van twee cijfers is een **controlegetal** dat berekend wordt op basis van het getal n dat gevormd wordt door de negen voorafgaande cijfers. Voor inwoners die geboren zijn vanaf het jaar 2000 moet er nog een extra cijfer 2 voor de negen cijfers gezet worden (+ 200000000) om het getal n te vormen. De berekening bestaat er dan in om n te delen door 97 (gehele deling) en de rest van die deling (modulo) af te trekken van 97. Dat verschil is het controlegetal. Het controlegetal wordt weergegeven met twee cijfers door eventueel een voorlooptul toe te voegen.

Op die manier weten we bijvoorbeeld dat het rijksregisternummer 93051822361 toegekend werd aan een man (oneven dagteller) die geboren is op 18 mei 1993 (eerste groep). We kunnen afleiden dat het geboortjaar 1993 is en niet 2093 (mochten we al zover zijn) omdat de deling van 930518223 door 97 rest 36 oplevert. We krijgen dan dat $97 - 36$ gelijk is aan het controlegetal 61. Als de inwoner geboren was in 2093 dan zouden we voor de berekening van het controlegetal 2930518223 delen door 97 om rest 7 te bekomen. In dat geval had het controlecijfer $97 - 7 = 90$ moeten zijn.

Opgave

Schrijf een `bash` shell script `rrn` dat gebruikt kan worden om de geldigheid van een gegeven rijksregisternummer te controleren en om er informatie uit af te leiden. Het rijksregisternummer moet als eerste argument aan het shell script doorgegeven worden. Alle andere argumenten worden genegeerd. Als er geen argumenten doorgegeven worden dan is het gegeven rijksregisternummer gelijk aan de lege string.

Het shell script moet achtereenvolgens controleren of het gegeven rijksregisternummer aan de volgende voorwaarden voldoet:

- Het rijksregisternummer moet exact 11 cijfers bevatten (alle karakters die geen cijfer zijn worden genegeerd). Als dat niet het geval is dan moet de volgende boodschap uitgeschreven worden naar `stderr` en moet het shell script eindigen met *exit status 1*.

```
rrn: ongeldig formaat (n cijfers)
```

Hierbij is n het aantal cijfers in het rijksregisternummer. Als $n = 1$ dan moet de boodschap de enkelvoudsvorm (1 cijfer) gebruiken.

- Het getal dat gevormd wordt door de laatste twee cijfers van het rijksregisternummer moet gelijk zijn aan één van de twee mogelijke controlegetallen die berekend worden op basis van de eerste negen cijfers. Als dat niet het geval is dan moet de volgende boodschap uitgeschreven worden naar `stderr` en moet het shell script eindigen met *exit status 2*.

```
rrn: ongeldig controlegetal
```

- We gaan ervan uit dat rijksregisternummers waarvan de laatste twee cijfers gelijk zijn aan het berekende controlegetal voor inwoners geboren vanaf het jaar 2000 corresponderen met inwoners die geboren zijn in de 21ste eeuw (20xx) en de andere met inwoners die geboren zijn in de 20ste eeuw (19xx). Als het rijksregisternummer geen geldige geboortedatum bevat dan moet de volgende boodschap uitgeschreven worden naar `stderr` en moet het shell script eindigen met *exit status 3*.

```
rrn: ongeldige geboortedatum
```

Als aan alle voorwaarden voldaan is dan moet het shell script de volgende informatie over het rijksregisternummer uitschrijven naar standaard uitvoer en eindigen met *exit status 0*.

```
rijksregisternummer:  jj.mm.dd-xxx.cc  
geslacht:  geslacht  
geboortedatum:  dd/mm/jjjj
```

Hierbij moeten de cursieve fragmenten ingevuld worden met informatie die afgeleid wordt uit het gegeven rijksregisternummer:

- De cijfers *jj*, *mm* en *dd* corresponderen met de geboortedatum (respectievelijk jaar, dag en maand), *xxx* met de dagteller en *cc* met het controlegetal zoals ze voorkomen in het rijksregisternummer.
- De string *geslacht* correspondeert met het geslacht dat kan afgeleid worden uit de dagteller: *man* voor mannen en *vrouw* voor vrouwen.
- De cijfers *dd*, *mm* en *jjjj* corresponderen met de geboortedatum (respectievelijk dag, maand en jaar), waarbij het jaartal volledig moet uitgeschreven worden (vier cijfers).

Voorbeeld

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

```
$ rrn "75.12.05-137.14"
rijksregisternummer: 75.12.05-137.14
geslacht: man
geboortedatum: 05/12/1975
$ echo $?
0
$ rrn ">>>09082428248<<<LENA<NADINE<INGRID"
rijksregisternummer: 09.08.24-282.48
geslacht: vrouw
geboortedatum: 24/08/2009
$ echo $?
0
$ rrn 1234567890
rrn: ongeldig formaat (10 cijfers)
$ echo $?
1
$ rrn xxx3xxx
rrn: ongeldig formaat (1 cijfer)
$ echo $?
1
$ rrn
rrn: ongeldig formaat (0 cijfers)
$ echo $?
1
$ rrn 85.02.01-002.00
rrn: ongeldig controlegetal
$ echo $?
2
$ rrn 85.02.30-002.24
rrn: ongeldige geboortedatum
$ echo $?
3
```

Opgave 4 (running cargo)

(10 pt)

Onderstaande paragrafen zijn afkomstig uit Rudyard Kipling's verhaal *Brother Square-Toes* dat dateert van 1910. Wat valt er op aan de vetgedrukte passage?

*"I'll have to bide ashore and grow cabbages for a while, after I've run this cargo; but I do wish-- Dad says, going over the **lugger's side with our New Year presents under his arm and young L'Estrange holding the lantern** — "I just do wish that those folk which make war so easy had to run one cargo a month all this winter. It 'ud show 'em what honest work means."*

"Well, I've warned ye," says Uncle Aurette. "I'll be slipping off now before your Revenue cutter comes. Give my love to Sister and take care o' the kegs. It's thicking to southward." 'I remember him waving to us and young Stephen L'Estrange blowing out the lantern. By the time we'd fished up the kegs the fog came down so thick Dad judged it risky for me to row 'em ashore, even though we could hear the ponies stamping on the beach. So he and Uncle Lot took the dinghy and left me in the smack playing on my fiddle to guide 'em back.

De passage bevat geen enkel woord dat meermaals voorkomt. Met 66 woorden is het de langste passage met die eigenschap die wiskundige Mike Keith ontdekte in een onderzoek uit 2002 van alle manuscripten op Project Gutenberg.

Vreemd genoeg komt er in hetzelfde verhaal nog een tweede uitzonderlijke passage voor — 15 opeenvolgende woorden van drie letters.

*"What?" says Toby, "I thought it was Gert Schwankfelder." He put down his fiddle and took a good look at me. "Himmel!" he says. "I have hit the wrong boy. It is **not the new boy**. Why are you not the new boy? Why are you not Gert Schwankfelder?"*

Opgave

Schrijf een `bash` shell script `highlight` dat kan gebruikt worden om een bepaalde passage uit een tekstbestand in het vet weer te geven. Het tekstbestand wordt ingelezen via standaard invoer. Aan het shell script moeten twee argumenten $r_1:k_1$ en $r_2:k_2$ doorgegeven worden, waarbij $r_i, k_i \in \mathbb{N}_0$ ($i = 1, 2$). De passage die in het vet moet weergegeven worden, begint bij het k_1 -de karakter op de r_1 -de regel en eindigt bij het k_2 -de karakter op de r_2 -de regel (grenzen inbegrepen). Hierbij worden zowel de posities van de regels als de posities van de karakters op een regel genummerd vanaf 1, respectievelijk van boven naar onder en van links naar rechts. Als het aantal karakters op een regel kleiner is dan k_1 dan start de passage na het laatste karakter op de regel. Als het aantal karakters op een regel kleiner is dan k_2 dan eindigt de passage bij het laatste karakter van de regel.

ANSI escape sequences

Maak gebruik van zogenaamde ANSI *escape sequences* om een passage in het vet weer te geven op de Unix commandolijn.

```
$ echo -e 'Dit wordt in het \033[1mVET\033[0m weergegeven!'
Dit wordt in het VET weergegeven!
```

Hierbij zet de *escape sequence* `\033[1m` de weergave van vette tekst aan en zet de *escape sequence* `\033[0m` de weergave van vette tekst terug uit. Let ook op het gebruik van de optie `-e` bij het commando `echo`. Deze optie zorgt ervoor dat backslashes geïnterpreteerd worden als escape-karakters.

Het shell script moet normaalgezien enkel de regels van de gegeven passage uitschrijven naar standaard uitvoer. Het script moet echter ook de volgende opties ondersteunen, waaraan telkens een natuurlijk getal n als verplicht argument moet meegegeven worden:

- optie `-a <int>`: schrijf ook de n regels die volgen op de gegeven passage uit; als er minder dan n regels volgen op de passage, dan moeten die allemaal uitgeschreven worden
- optie `-b <int>`: schrijf ook de n regels die voorafgaan aan de gegeven passage uit; als er minder dan n regels voorafgaan aan de passage, dan moeten die allemaal uitgeschreven worden

Het shell script moet voor de verwerking van de opties de flexibiliteit aan de dag leggen die gebruikelijk is bij Unix commando's: volgorde van opties speelt geen rol, opties kunnen eventueel samengenomen worden, Het shell script moet er ook voor zorgen dat de weergave van vette tekst altijd uitstaat nadat het shell script werd uitgevoerd, ook al overlapt het aangegeven interval niet of slechts gedeeltelijk met de regels in het tekstbestand. Het shell script moet ook de volgende foutafhandeling voorzien:

- als het shell script niet de gepaste opties meekrijgt (enkel ondersteuning voor de opties `-a` en `-b` met een natuurlijk getal als verplicht argument) of als er niet juist twee argumenten worden doorgegeven van de vorm $r:k$ met $r, k \in \mathbb{N}_0$, dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stderr` en moet het shell script eindigen met `exit status 1`
- als de gegeven beginpositie $r_1 : k_1$ (eerste argument) na de gegeven eindpositie $r_2 : k_2$ (tweede argument) ligt, dan moet de gepaste boodschap (zie onderstaande voorbeeldsessie) uitgeschreven worden naar `stderr` en moet het shell script eindigen met `exit status 2`; de twee posities mogen wel gelijk zijn, wat aangeeft dat er één regel moet uitgeschreven worden waarop één enkel karakter in het vet wordt aangeduid

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

Voorbeeld

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

```
$ echo "12345678901234567890" | highlight 1:5 1:12
12345678901234567890
$ echo $?
0
$ echo "12345678901234567890" | highlight 1-5 1-12
Syntax: highlight [-a <int>] [-b <int>] <int>:<int> <int>:<int>
$ echo $?
1
$ echo "12345678901234567890" | highlight 1:5
Syntax: highlight [-a <int>] [-b <int>] <int>:<int> <int>:<int>
$ echo $?
1
$ echo "12345678901234567890" | highlight -x 1:5 1:12
Syntax: highlight [-a <int>] [-b <int>] <int>:<int> <int>:<int>
$ echo $?
1
$ echo "12345678901234567890" | highlight -a 1:5 1:12
Syntax: highlight [-a <int>] [-b <int>] <int>:<int> <int>:<int>
$ echo $?
1
$ echo "12345678901234567890" | highlight 1:5 1:5
12345678901234567890
$ echo "12345678901234567890" | highlight 1:12 1:5
highlight: invalid interval
$ echo $?
2
$ echo "12345678901234567890" | highlight 1:0 1:12
highlight: invalid interval
$ echo $?
2
$ cat kipling.txt | highlight 4035:22 4036:11
the wrong boy. It is not the new boy. Why are you not the new boy? Why
are you not Gert Schwankfelder?
```



```
$ cat kipling.txt | highlight -b2 4035:22 4036:11
```

What? says Toby, I thought it was Gert Schwankfelder. He put down his fiddle and took a good look at me. Himmel! he says. I have hit the wrong boy. It is not the new boy. Why are you not the new boy? Why are you not Gert Schwankfelder?

```
$ cat kipling.txt | highlight -a4 4035:22 4036:11
```

the wrong boy. It is not the new boy. Why are you not the new boy? Why are you not Gert Schwankfelder?

I dont know, I said. The gentleman in the pink blanket brought me.

Says the Indian, He is hungry, Toby. Christians always feed the

```
$ cat kipling.txt | highlight -b2 -a4 4035:22 4036:11
```

What? says Toby, I thought it was Gert Schwankfelder. He put down his fiddle and took a good look at me. Himmel! he says. I have hit the wrong boy. It is not the new boy. Why are you not the new boy? Why are you not Gert Schwankfelder?

I dont know, I said. The gentleman in the pink blanket brought me.

Says the Indian, He is hungry, Toby. Christians always feed the

```
$ cat kipling.txt | highlight 3886:54 3893:46
```

this cargo; but I do wish--Dad says, going over the luggers side with our New Year presents under his arm and young LEstrange holding the lantern--I just do wish that those folk which make war so easy had to run one cargo a month all this winter. It ud show em what honest work means.

Well, Ive warned ye, says Uncle Aurette. Ill be slipping off now before your Revenue cutter comes. Give my love to Sister and take care

```
$ cat kipling.txt | highlight -b1 -a1 3886:54 3893:46
```

Ill have to bide ashore and grow cabbages for a while, after Ive run this cargo; but I do wish--Dad says, going over the luggers side with our New Year presents under his arm and young LEstrange holding the lantern--I just do wish that those folk which make war so easy had to run one cargo a month all this winter. It ud show em what honest work means.

Well, Ive warned ye, says Uncle Aurette. Ill be slipping off now before your Revenue cutter comes. Give my love to Sister and take care o the kegs. Its thicking to southward. I remember him waving to us