
EXAMEN: Scriptingtalen

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

maandag 22-08-2011, 14:00
academiejaar 2010-2011
tweede zittijd

Opgave 1

Een **tekstmarkeertaal** — kortweg markeertaal of *markup language* — is een computertaal om tekstdocumenten te voorzien van aanwijzingen ten behoeve van de softwarematige verwerking. Deze aanwijzingen worden in de tekst zelf ingevoegd. Een **opmaaktaal** is een tekstmarkeertaal waarbij de aanwijzingen instructies zijn voor de opmaak bij de weergave van de tekst. Hierbij worden speciale sleutelwoorden gebruikt om de weergave van lettertypes, pagina's, paragrafen, enz., te sturen. L^AT_EX, troff en HTML zijn voorbeelden van opmaaktalen.

Software voor spellingscontrole gaat vaak de mist in wanneer ze als dusdanig wordt toegepast op een tekstbestand dat aanwijzingen uit een opmaaktaal bevat. Vaak is het nodig om extra filters te schrijven of de spellingscontrolesoftware zelf aan te passen, zodat opmaakaanwijzingen herkend en genegeerd worden, en de spellingscontrole enkel wordt uitgevoerd op de letterlijke tekst. Voor deze opgave vragen we je om een filter te schrijven die de aanwijzingen van een eenvoudige opmaaktaal uit een tekstbestand verwijderd en enkel de letterlijke tekst naar standaard uitvoer uitschrijft.

De opmaaktaal die we voor ogen hebben, is er een die toelaat om de weergave van lettertypes in een tekst te wijzigen. Elke aanwijzing wordt voorafgegaan door een backslash (\). Indien het karakter dat onmiddellijk volgt op een backslash geen commando is uit onderstaande lijst, dan wordt dat karakter uitgeschreven als onderdeel van de letterlijke tekst. Zo kan bijvoorbeeld de opmaak \\ gebruikt worden om één enkele backslash () uit te schrijven.

- \b aan- of uitschakelen van vette tekst (initiële toestand is uit)
- \i aan- of uitschakelen van cursieve tekst (initiële toestand is uit)
- \s instellen van lettergrootte; het commando \s wordt optioneel gevolgd door een getal dat de grootte van het lettertype aangeeft; indien er geen getal volgt op het commando \s, dan stelt het commando opnieuw de voorgaande lettergrootte in; het getal dat volgt op het commando \s kan een decimaal punt bevatten: 12, 9.5, 11. en .5 moeten allemaal herkend worden als geldige getallen
- * aan- of uitschakelen van de verwerking van opmaakaanwijzingen (initiële toestand is aan); wanneer de verwerking is uitgeschakeld, worden de opmaakaanwijzingen als letterlijke tekst beschouwd; voor de eenvoud mag je ervan uitgaan dat elk commando * dat de verwerking uitschakelt, op dezelfde regel terug zal worden ingeschakeld door een ander commando *

Gevraagd wordt om een sed script `verwijder_opmaak.sed` te schrijven, dat de opmaakaanwijzingen kan wegfilteren van de eenvoudige opmaaktaal die hierboven werd beschreven. Onderstaande sessie illustreert hoe dit script moet kunnen gebruikt worden.

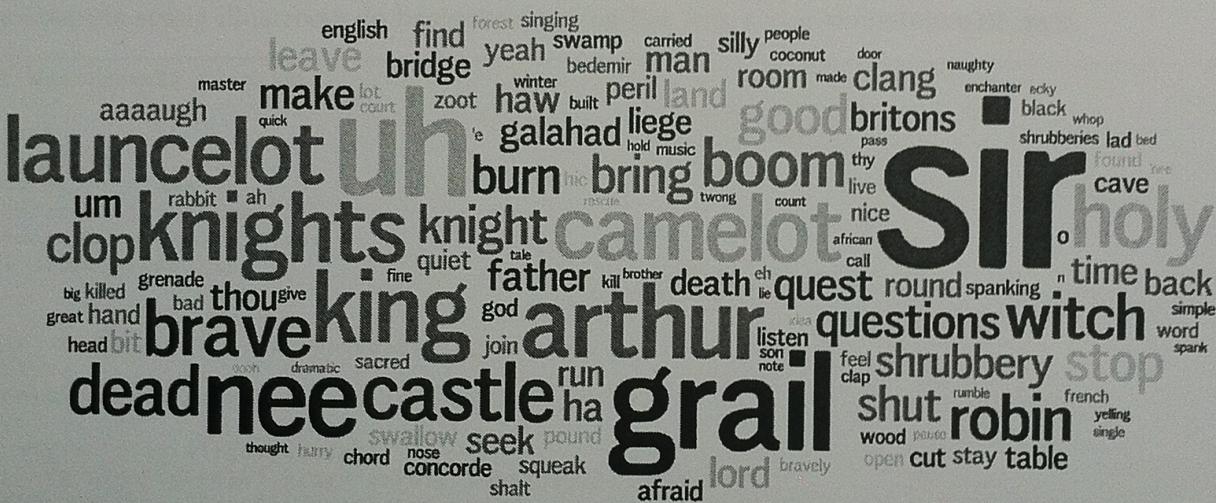
```
$ cat tekst_met_opmaak.txt
\s18.\bOPMAAK voorbeeld\b\s
/*Gebruik het commando \b om vette tekst weer te geven.*/
Gebruik het commando \i als je iets wilt \ibenadrukken\i.
Gebruik een \s14GROOT\s lettertype voor titels: doorgaans volstaat 14 punten.
Onthoud dat alle commando's schakelen, behalve het commando \s.
Er kan \*ook \bmeerdere\b keren\* \iper\i regel \*\bgeschakeld\b worden.*/
$ sed -f verwijder_opmaak.sed tekst_met_opmaak.txt
OPMAAK voorbeeld
```

Gebruik het commando \b om vette tekst weer te geven.
Gebruik het commando \i als je iets wilt benadrukken.
Gebruik een GR0OT lettertype voor titels: doorgaans volstaat 14 punten.
Onthoud dat alle commando's schakelen, behalve het commando \s.
Er kan ook \bmeerdere\b keren per regel \bgeschakeld\b worden.

Opmerking: Probeer zo weinig mogelijk bijkomende veronderstellingen te maken over de tekst waarin de opmaakaanwijzingen werden aangebracht. Geef duidelijk aan via commentaar in je script of op je antwoordblad indien je toch bijkomende veronderstellingen maakt (bijvoorbeeld als je ervan uitgaat dat bepaalde karakters niet in de tekst voorkomen).

Opgave 2

Vandaag de dag zie je om de haverklap wel ergens een woordenwolk (*tag cloud*) opduiken op het Internet. Dit is een visuele weergave van de frequentie waarmee woorden in een tekst voorkomen. Woorden die vaker voorkomen worden in een groter lettertype weergegeven. Op die manier springen de belangrijkste woorden van de tekst direct in het oog. Het visuele effect kan nog versterkt worden door te werken met verschillende kleuren en plaatsing van de woorden. Meestal wordt een lijst van stopwoorden gebruikt. Woorden die in deze lijst voorkomen worden niet weergegeven in de woordenwolk.



Gevraagd wordt om een awk script `woordenwolk.awk` te schrijven, dat een woordenwolk opstelt voor een gegeven tekst. Het awk script moet hiervoor als volgt te werk gaan:

1. Lees de tekst uit standaard invoer, breek die op in afzonderlijke woorden en tel het aantal keer dat elk woord in de tekst voorkomt. De woorden uit de tekst worden gevormd door de langst mogelijke opeenvolging van letters en de apostrof ('). Voor de telling worden alle woorden omgezet naar kleine letters, zodat geen onderscheid gemaakt wordt tussen hoofdletters en kleine letters. Het tekstfragment "Don't say that word!" bestaat dus uit de vier woorden don't, say, that en word.
 2. In de huidige directory staat een tekstbestand **stopwoorden.txt** dat een lijst van woorden bevat. Hierbij staat elk woord op een afzonderlijke regel. De woorden die in deze lijst voorkomen moeten niet opgenomen worden in de woordenwolk.

Opmerking: de lijst van stopwoorden mag niet vast ingebakken zitten in het **awk** script, maar moet uit het stopwoordenbestand gelezen worden.

 3. In de huidige directory staat een HTML-bestand **woordenwolk.html** dat als sjabloon voor de weergave van een woordenwolk in een browser moet gebruikt worden. Als je de inhoud van dit

Opmerking: de lijst van stopwoorden mag niet vast ingebakken zitten in het awk script, maar moet uit het stopwoordenbestand gelezen worden.

HTML-bestand bekijkt, dan zal je merken dat daarin het volgende fragment voorkomt:

```
<param name="wordcounts" value="" />
```

Je kunt een woordenwolk laten tekenen voor de gegeven tekst, door een gepaste waarde toe te kennen aan het `value` attribuut. Deze waarde is een string die een reeks van naam-waarde paren bevat, waarbij de verschillende paren van elkaar worden gescheiden door een komma (,), en de naam wordt gescheiden van de waarde door een dubbelpunt (:). De naam is telkens een woord dat in de tekst voorkomt, en de waarde is het corresponderend aantal keer dat het woord voorkomt. De volgorde waarin de naam-waarde paren worden opgeliist, speelt geen rol bij de weergave.

Veronderstel bijvoorbeeld dat de woorden `een`, `twee`, ..., `zes` elk respectievelijk één keer, twee keer, ..., zes keer voorkomen in de gegeven tekst. Veronderstel voorts dat de woorden `een` en `drie` stopwoorden zijn. Dan moet de waarde van het fragment uit het HTML-bestand op de volgende manier ingevuld worden:

```
<param name="wordcounts" value="twee:2,vier:4,vijf:5,zes:6" />
```

Opmerking: het sjabloon mag niet vast ingebakken zitten in het `awk` script, maar moet uit het HTML-bestand gelezen worden.

Onderstaande sessie illustreert de basiswerking van het `awk` script `woordenwolk.awk`.

```
$ cat woordenwolk_test.txt
zes
vijf,zes
vier,vijf,zes
drie,vier,vijf,zes
twee,drie,vier,vijf,zes
een,twee,drie,vier,vijf,zes
$ cat stopwoorden.txt
een
drie
$ gawk -f woordenwolk.awk woordenwolk_test.txt
<html><title>woordenwolk</title></head>
<body>
<div style="text-align:center">
  <applet name="wordle" code="wordle.WordleApplet.class"
    codebase="http://wordle.appspot.com" archive="/j/v1354/wordle.jar" width="800" height="600">
    <param name="wordcounts" value="vijf:5,vier:4,zes:6,twee:2" />
    Het ziet ernaar uit dat je browser het element APPLET niet kent.
    Je moet de <a href="http://java.com/">Java</a> plugin installeren en inschakelen.
  </applet>
</div>
</body>
</html>
$
```

Opmerking: Bij dit examen worden de bestanden `stopwoorden.txt` en `woordenwolk.html` meegeleverd. Als je je `awk` script uittest op het tekstbestand `monty_python_holy_grail.txt` — dat eveneens bij het examen werd meegeleverd — dan moet dit de weergave van een woordenwolk opleveren zoals in bovenstaande afbeelding.

Opgave 3

Schrijf een `bash` shell script `wwgen` dat p wachtwoorden van lengte l genereert, en een evaluatie maakt van de sterkte van elk van deze wachtwoorden. Dit shell script moet als volgt op de commandolijn

kunnen aangeroepen worden:

`wwgen [-c] [-p aantal] [-l lengte] patroon bestand...`

Hierbij mogen de opties in willekeurige volgorde opgegeven worden, en moeten ze op de gebruikelijke manier kunnen samengenomen worden (bv. de reeks opties `-c -p 3` mag ook als `-cp3` geschreven worden). De vaste argumenten van het shell script zijn dus een patroon en één of meer bestanden, waarvan de betekenis verder wordt uitgelegd.

Het shell script moet een wachtwoord genereren door willekeurig l karakters te selecteren uit een string die de volgende karakters bevat:

```
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789,.!@#$%^&*()[]{};:_+=
```

Hetzelfde karakter mag hierbij verschillende keren gekozen worden. Het genereren van wachtwoorden wordt verder gestuurd door de volgende opties:

- `-c` met deze optie bevatten de wachtwoorden geen kleine letters; het wachtwoord mag dus enkel bestaan uit hoofdletters, cijfers en leestekens
- `-l n` gegenereerde wachtwoorden hebben lengte n ; indien de lengte niet wordt ingesteld via deze optie, zijn wachtwoorden even lang als het patroon dat als argument wordt meegegeven
- `-p n` er worden n wachtwoorden gegenereerd; standaard worden er 5 wachtwoorden gegeneerd indien deze optie niet wordt meegegeven

Om de sterkte van het wachtwoord te bepalen, wordt eerst een score berekend door telkens 1 punt toe te kennen voor elk van de volgende criteria waaraan het wachtwoord voldoet:

- aantal hoofdletters en kleine letters samen \geq aantal keer dat letter L voorkomt in patroon¹
- aantal cijfers \geq aantal keer dat letter C voorkomt in patroon
- aantal leestekens (geen letter of cijfer) \geq aantal keer dat letter T voorkomt in patroon
- wachtwoord komt niet voor in de bestanden die als argument worden meegegeven; hierbij wordt verondersteld dat deze bestanden een lijst van woorden bevatten, die elk op een afzonderlijke regel staan

De onderstaande tabel geeft de evaluatie van de sterkte van een wachtwoord aan, op basis van de score die voor het wachtwoord wordt berekend:

SCORE	EVALUATIE
0	zeer zwak
1	zwak
2	normaal
3	sterk
4	zeer sterk

Het shell script moet de volgende controles uitvoeren op de informatie die via de commandolijn wordt doorgegeven. Indien geen correcte informatie werd meegegeven, moet een passende foutbericht naar standaard error geschreven worden:

¹Het patroon is een string die enkel bestaat uit de letters L, C en T. De volgorde van de letters in het patroon is niet belangrijk. Het patroon wordt zowel gebruikt om de standaardlengte van de wachtwoorden vast te leggen, alsook om te bepalen hoeveel letter, cijfers en leestekens het wachtwoord minimaal moet bevatten.

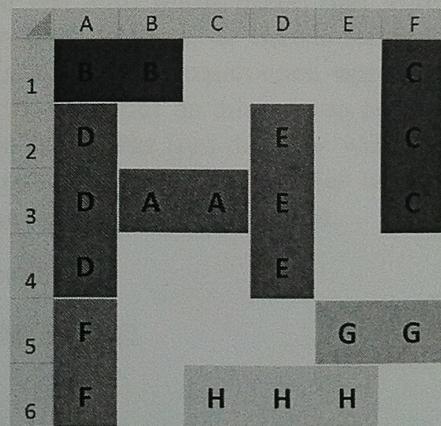
- aan optie -p moet een getal groter of gelijk aan 1 als argument doorgegeven worden
- aan optie -l moet een getal als argument doorgegeven worden dat minstens even groot is als de lengte van het patroon
- er mogen geen andere opties meegegeven worden dan de opties -c, -l en -p
- het eerste argument moet een patroon voorstellen dat enkel bestaat uit de letters L, C en T
- er moet minstens één bestand als argument worden doorgegeven, en alle doorgeven bestanden moeten bestaan en leesbaar zijn voor het shell script

Onderstaande sessie illustreert de basiswerking van het bash shell script wwggen, en geeft ook aan hoe de uitvoer er moet uitzien.

```
$ wwggen LLLLLCCTT /usr/dict/words
)aFs@M=jT sterk
9YL}uE&Os zeer sterk
sW.g7534H normaal
LgM(RjX:U sterk
$pyoR!ffB sterk
$ wwggen -l 15 -p 4 -c LCLCLTLT /usr/dict/words
-ZREU6IZ4!JPUU* zeer sterk
"WDGWS)OFOT:D## sterk
C(OUU@BQ%%C\={9 sterk
RHE9Y]:LOB*+3Y. zeer sterk
$ wwggen -cp3 -l12 CCLLLLTT /usr/dict/words blacklist.txt
{"IWB&Z_CXEZ sterk
OUZUNN8QM,:Y sterk
AL2MQ;@RO_2{ zeer sterk
$
```

Opgave 4

Bij uitzonderlijke omstandigheden (weer, ongevallen, enz.) kan het verkeer dusdanig in de war raken dat automobilisten nauwelijks nog vooruit of achteruit kunnen komen. In extreme gevallen waarbij weggebruikers elkaar indirect blokkeren, spreken we van een *verkeersinfarct*.



In deze opgave stellen we een verkeersinfarct voor binnen een rechthoekig gebied dat zich in de linkerbovenhoek van een Excel werkblad bevindt. De rechterfiguur hierboven geeft hiervan een voorbeeld. De verkeerssituatie stelt een aantal geblokkeerde auto's in een parkeergarage voor als rechthoekige blokken, die ofwel horizontaal of verticaal geparkeerd staan. De positie van elke auto wordt omschreven als een tekenreeks van vier karakters:

`<kolom> <rij> <richting> <lengte>`

Richting H duidt een auto aan die horizontaal geparkeerd staat, en die enkel naar links of naar rechts

kan bewegen indien hij daarbij niet geblokkeerd wordt door een andere auto. De eerste twee karakters staan respectievelijk voor de kolomletter en het rijnummer van de cel in Excel waar het linkerdeel van de auto geparkeerd staat. De lengte van de auto is minstens twee en geeft aan hoeveel horizontaal aangrenzende cellen hij beslaat. Zo stelt de tekenreeks B3H2 de wagen voor die in bovenstaande figuur met A gelabeld is.

Analoog duidt richting V een auto aan die verticaal geparkeerd staat. In dat geval kan de auto enkel naar boven en naar beneden bewegen, indien hij daarbij opnieuw niet gehinderd wordt door een andere auto. De eerste twee karakters geven nu de cel aan waar het bovenste deel van de auto geparkeerd staat. De lengte van de auto is minstens twee en geeft aan hoeveel verticaal aangrenzende cellen hij beslaat. Zo stelt de tekenreeks A2V3 de wagen voor die in bovenstaande figuur met D gelabeld is.

Gevraagd wordt:

- Schrijf een functie startSituatie in VBA met de volgende signatuur:

```
Function startSituatie(rijen As Integer, kolommen As Integer) As Worksheet
```

Deze functie moet achteraan de huidige werkmap een nieuw werkblad toevoegen, en een verwijzing naar dit werkblad als resultaat teruggeven. Alvorens het resultaat terug te geven moet de functie het werkblad ook nog opmaken zodat enkel een gegeven aantal rijen en kolommen in de linkerbovenhoek van het werkblad zichtbaar zijn. Voor de zichtbare cellen moet de breedte ingesteld worden op 5 en de hoogte op 30. De tekst in deze cellen moet zowel horizontaal als verticaal gecentreerd worden. Deze functie zal gebruikt worden om een verkeerssituatie te initialiseren.

- Schrijf een subprocedure tekenSituatie in VBA met de volgende signatuur:

```
Sub tekenSituatie(ws As Worksheet, rijen As Integer, kolommen As Integer, autos As String)
```

Deze subprocedure moet op het werkblad `ws` een verkeerssituatie uittekenen waarbij een aantal auto's geparkeerd staan binnen het rechthoekig gebied in de linkerbovenhoek van het werkblad. Het aantal rijen en kolommen van dit rechthoekig gebied wordt als parameter aan de subprocedure doorgegeven. De parameter `autos` omschrijft de positie van de geparkeerde auto's. Deze omschrijving is een tekenreeks met een lijst van posities van de individuele auto's, van elkaar gescheiden door een komma (,). De positie van een auto wordt omschreven zoals hierboven gedetailleerd werd uitgelegd. Indien een cel door een auto wordt bezet, dan wordt in deze cel de letter van de auto geschreven. De eerste auto heeft letter A, de tweede letter B, enzoverder. Omdat deze subprocedure verschillende keren na elkaar zal aangeroepen worden met verschillende verkeerssituaties, moet er ook voor gezorgd worden dat de inhoud van cellen waarop geen auto geparkeerd staat wordt ingevuld met de lege tekenreeks (om daarmee voorgaande labels te wissen). Bij wijze van voorbeeld geven we hieronder de weergave van het werkblad nadat de verkeerssituatie met de volgende omschrijving van auto's werd uitgetekend:

`B3H2,A1H2,F1V3,A2V3,D2V3,A5V2,E5H2,C6H3`

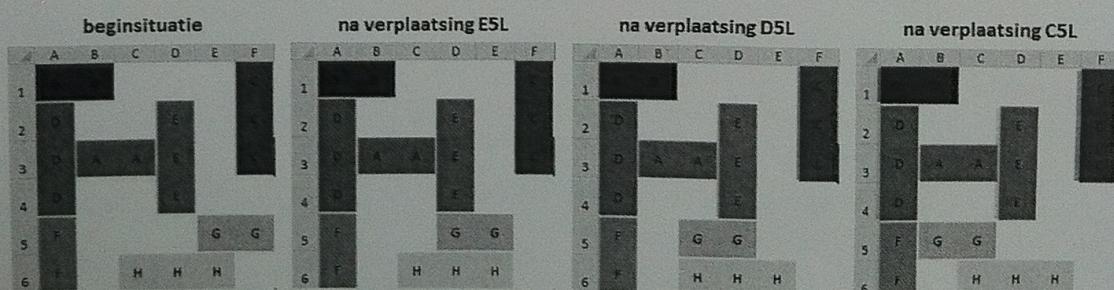
	A	B	C	D	E	F
1	B	B				C
2	D			E		C
3	D	A	A	E		C
4	D			E		
5	F				G	G
6	F		H	H	H	

Merk op dat dit de verkeerssituatie is die hierboven ook al stond afgebeeld. Daarbij werd de

achtergrond van cellen waarop wagens stonden geparkeerd ook nog eens extra ingekleurd. Het is niet nodig om je subprocedure naast de labels ook nog een achtergrondkleur te laten toekennen.

3. Schrijf een subprocedure **verkeersinfarct** in VBA, waaraan geen parameters moeten meegegeven worden. Deze subprocedure moet gebruik maken van de procedures **startSituatie** en **tekenSituatie** om visueel aan te geven hoe een verkeersinfarct kan ontbonden worden. Deze subprocedure moet achtereenvolgens de volgende stappen uitvoeren:

- Vraag via een invoervenster de naam op van een werkblad. Zorg ervoor dat de waarde die wordt ingevoerd in het invoervenster als standaardwaarde gebruikt wordt als de subprocedure **verkeersinfarct** opnieuw wordt uitgevoerd. Merk op dat de Excel werkmap **verkeersinfarct.xlsx** — die werd meegeleverd bij het examen — een reeks werkmappen **leveln** ($1 \leq n \leq 5$) bevat met gegevens over een verkeersinfarct en de stappen naar de ontbinding ervan. Bedoeling is dat de naam van één van deze werkmappen wordt ingevuld in het invoervenster.
- Lees achtereenvolgens de volgende gegevens uit de tweede kolom van het opgegeven werkblad: *i*) aantal rijen, *ii*) aantal kolommen, *iii*) lijst met posities van geparkeerde auto's (gescheiden door komma's), *iv*) uitgang van de parkeergarage en *v*) reeks verplaatsingen om auto A naar de uitgang te krijgen (zie verder).
- Schrijf de beginsituatie van het verkeersinfarct uit op een nieuw werkblad dat achteraan de huidige werkmap wordt toegevoegd. Deze beginsituatie wordt omschreven door de gegevens op de eerste drie rijen in de tweede kolom van het opgegeven werkblad.
- Om het verkeersinfarct te ontbinden, moet auto A naar de uitgang van de parkeergarage gemanoeuvreerd worden. Hierbij staat F3R bijvoorbeeld voor een uitgang die zich aan de rechterrand van cel F3 bevindt. Voor deze opgave doet de specifieke positie van de uitgang er verder weinig toe. Op de vijfde rij in de tweede kolom van het opgegeven werkblad staat een tekenreeks met de omschrijving hoe auto A naar de uitgang kan geloodst worden. Hierbij wordt elke verplaatsing omschreven door een reeks van drie letters, en worden de opeenvolgende verplaatsingen van elkaar gescheiden door een komma. De eerste twee karakters van een verplaatsing duiden de linkerpositie (voor horizontaal geparkeerde auto's) of bovenpositie (voor verticaal geparkeerde auto's) aan van de auto die verplaatst wordt. Het derde karakter geeft de richting van de verplaatsing aan: naar boven (B), naar rechts (R), naar onder (O) of naar links (L). Hierbij wordt een auto telkens één cel verplaatst. Verplaatst telkens een volgende auto op basis van de omschrijving in de lijst van verplaatsingen. Geef na elke verplaatsing de nieuwe verkeerssituatie weer op het opgegeven werkblad. Op die manier wordt de ontbinding van de verkeerssituatie doorheen de tijd gevisualiseerd op het werkblad. Hieronder krijg je alvast een weergave van de verkeerssituatie na een aantal opeenvolgende verplaatsingen:



Hint: Om een wachttijd van 1 seconde in te lassen tussen twee opeenvolgende verplaatsingen, kan je de volgende VBA instructie uitvoeren:

```
Call Application.Wait(Now + TimeValue("0:00:01"))
```