Nakijker View

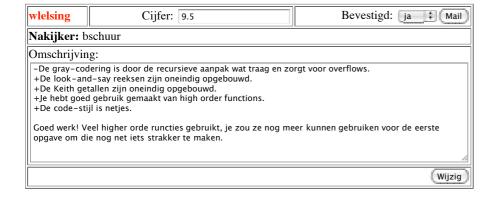
How to use this page: reviewer (studentassistent), lecturer

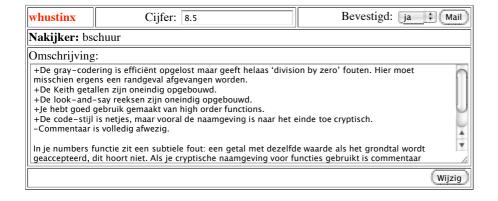
For lecturers: return to the assignments page

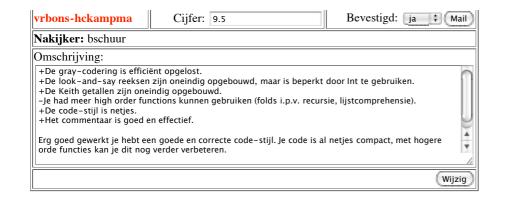
Opgave 1

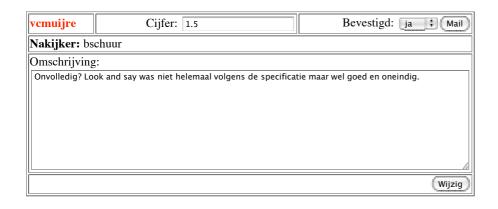


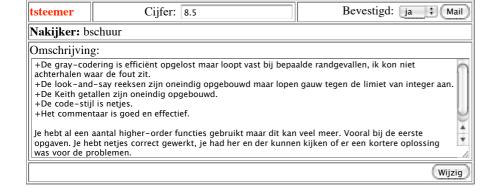








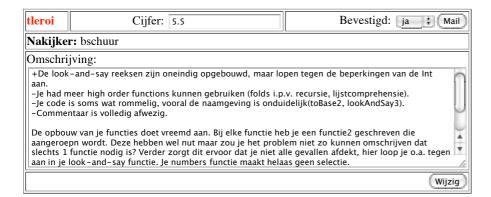


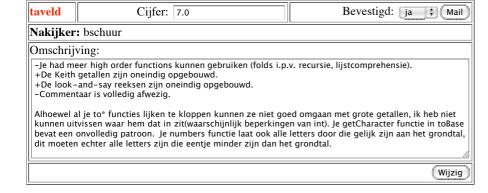


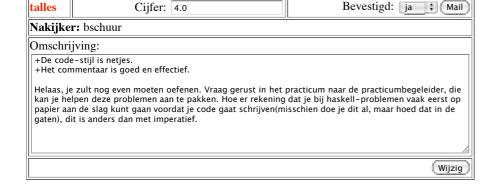




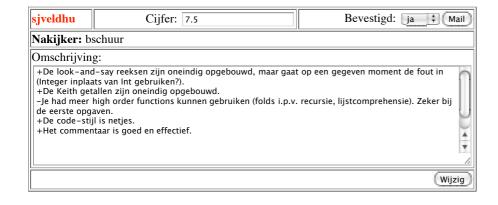


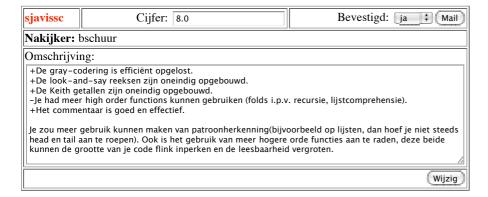


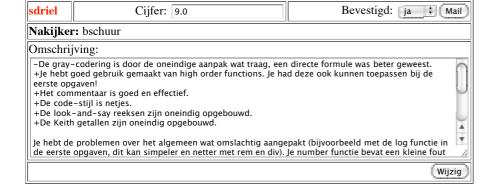




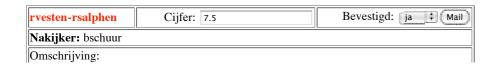
sttimmer	Cijfer:	10	Bevestigd: ja [Mail)			
Nakijker: bschuur						
Omschrijving	:					
+De Jook-and- +De Keith geta +Je hebt goed (+Je hebt goed (+De code-stijl +Het comment Supergoed werl	aar is goed en effection k! Proficiat! Netjes on	indig opgebouwd. gebouwd. lijstcomprehensies. high order functions. ef. n alle definities bovenaan in	de file te zetten. Je maakt zeer goed emen netjes en kloppend op.			
			(Wijzig)			



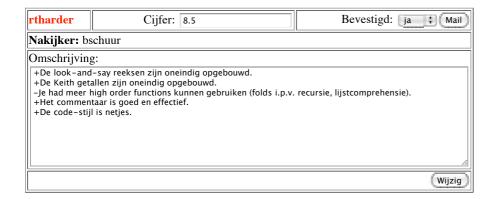


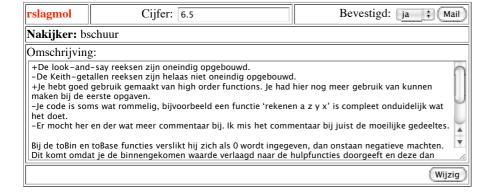


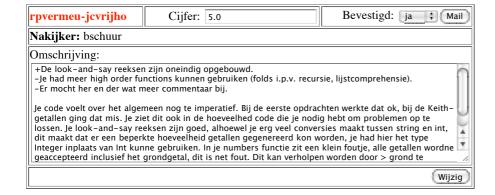


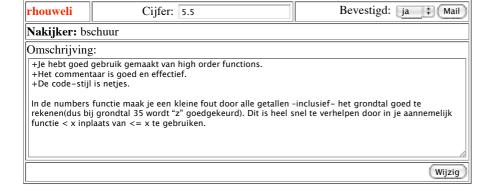


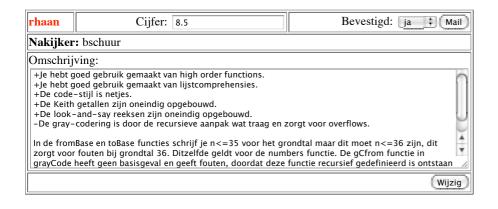


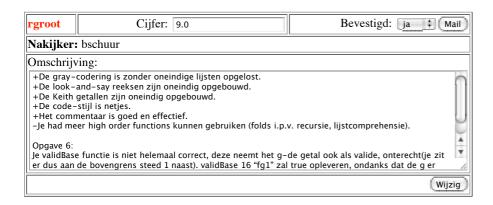


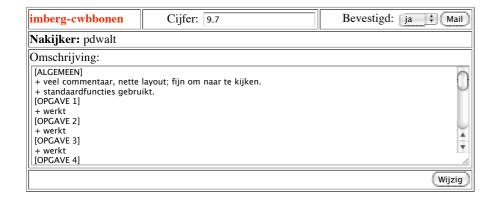


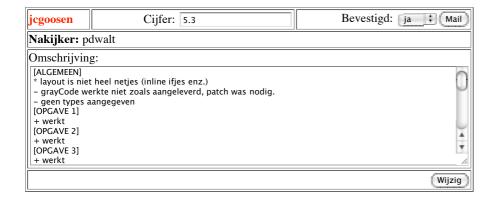




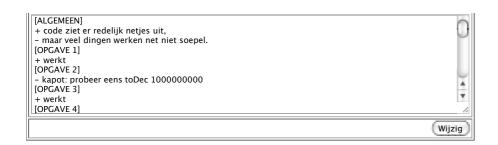


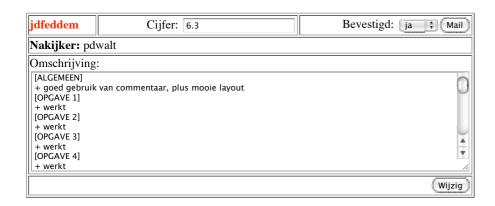


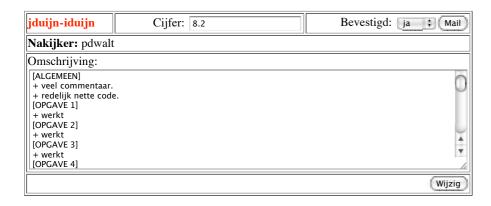


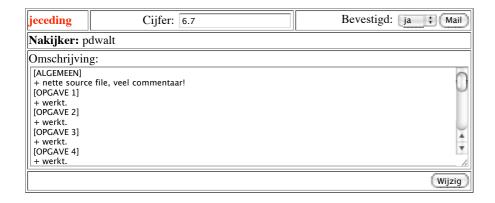












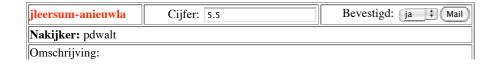




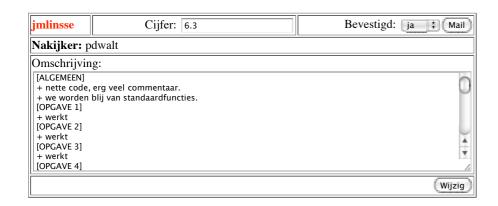


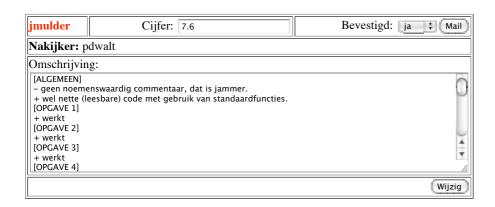




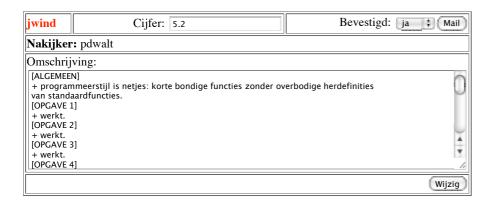


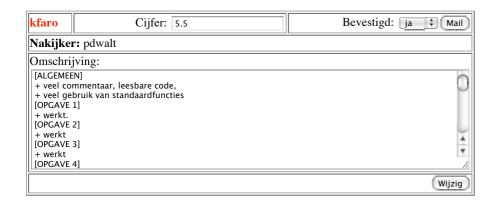






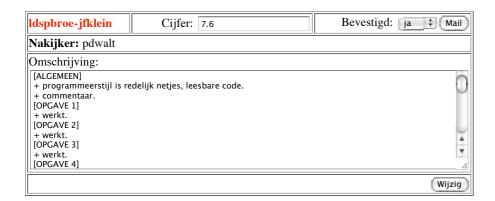




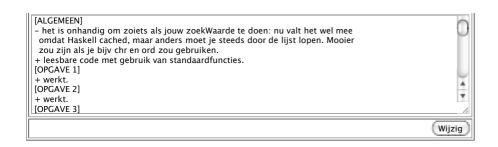


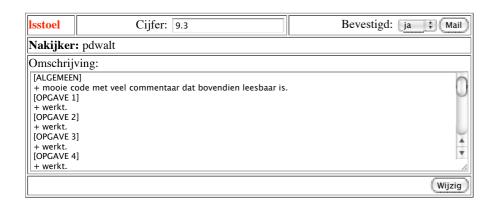


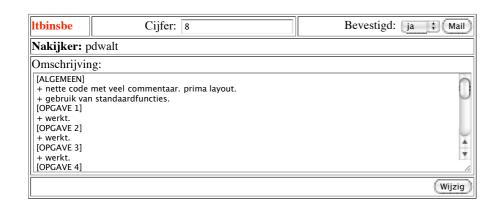




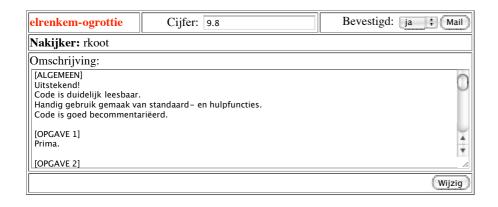


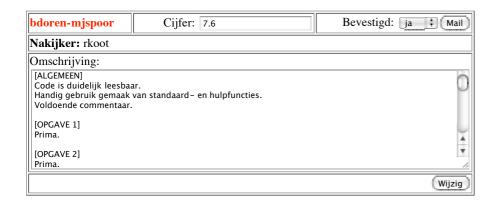


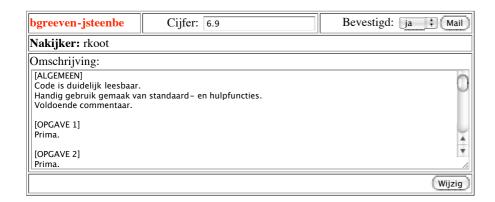


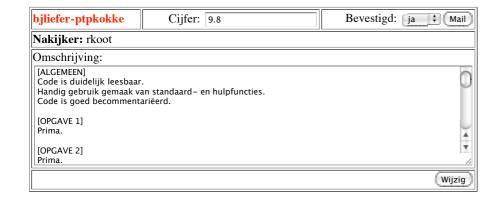


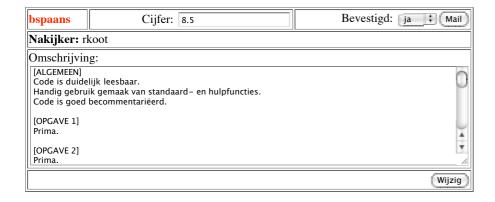






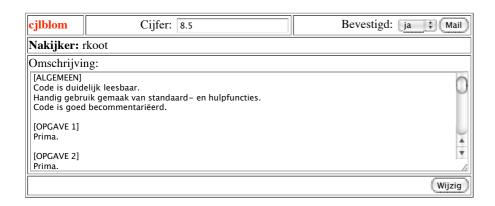


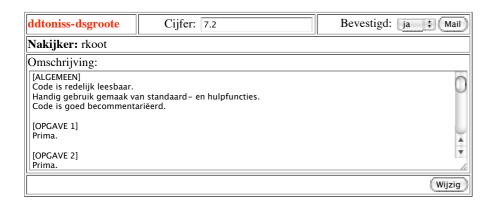


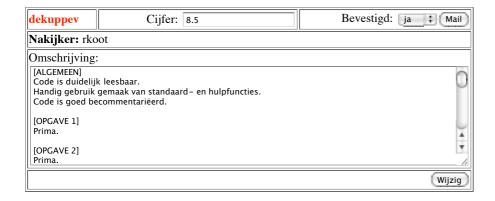


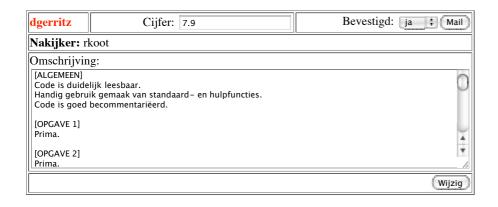


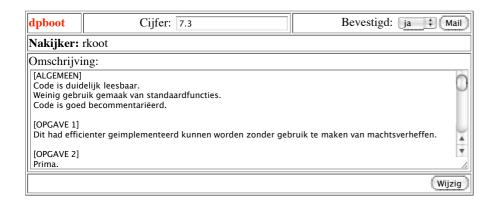


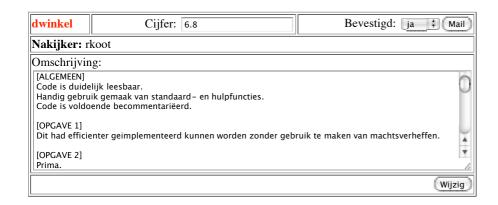


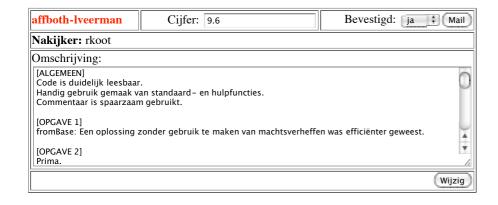


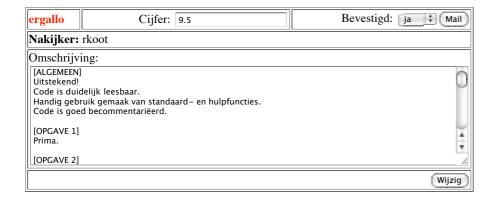






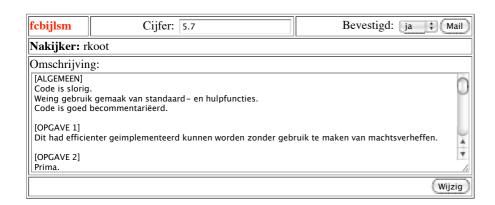


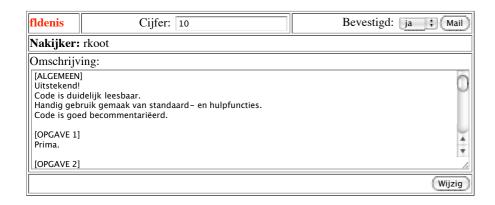


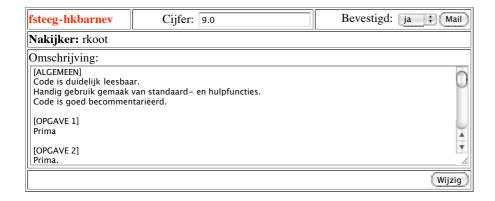


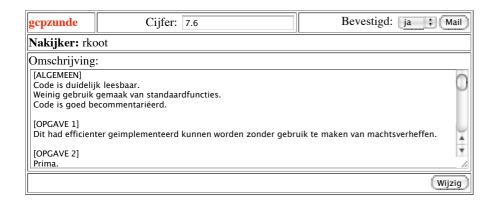




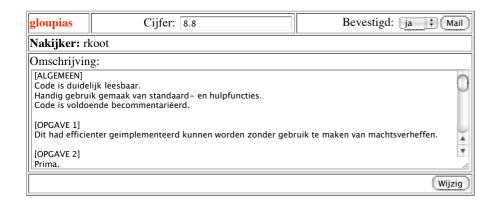


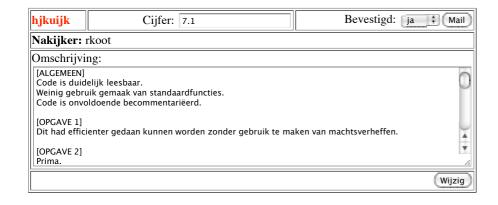


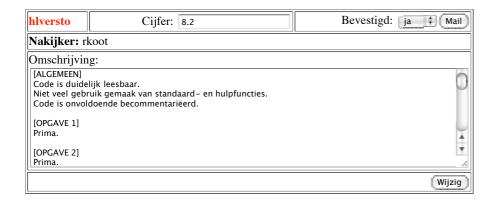






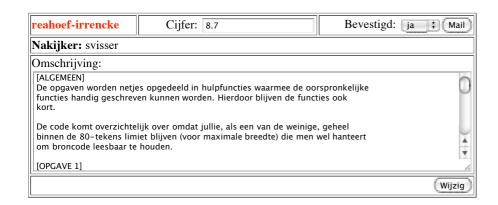


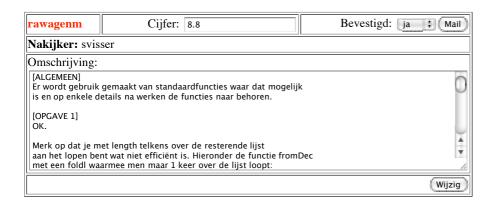


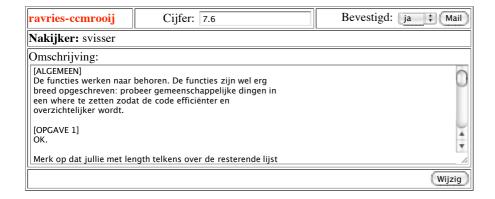






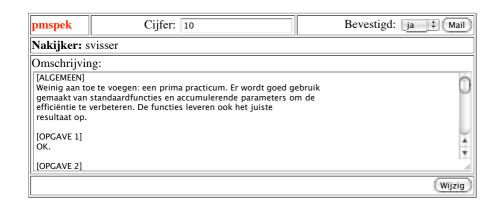


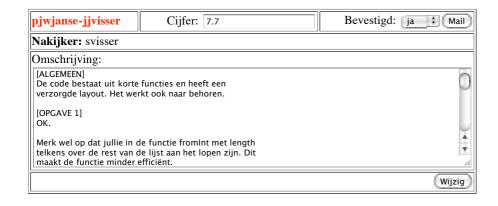


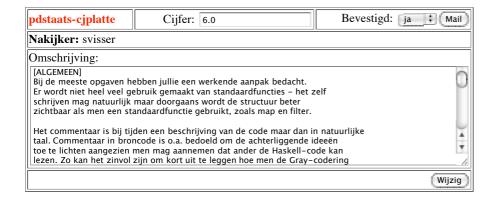




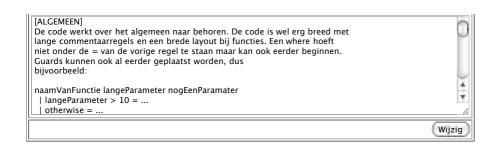


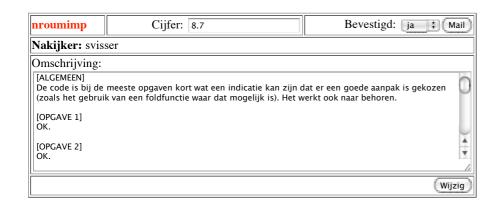


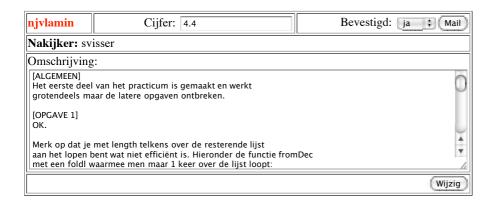


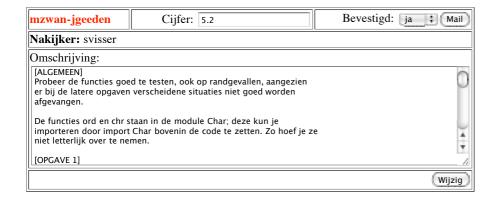


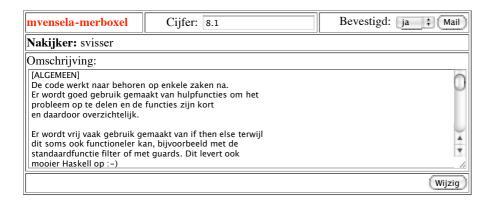


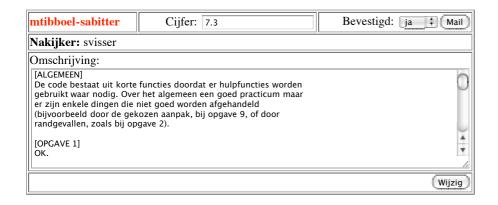




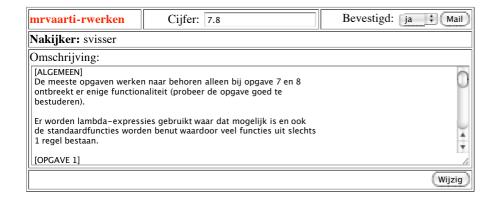


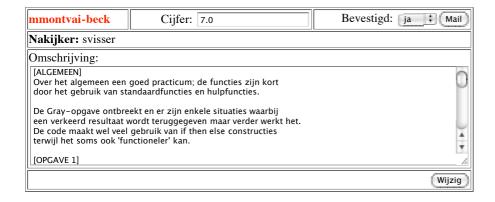


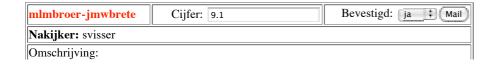




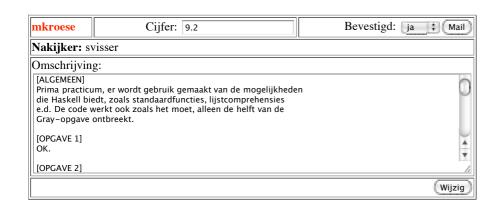




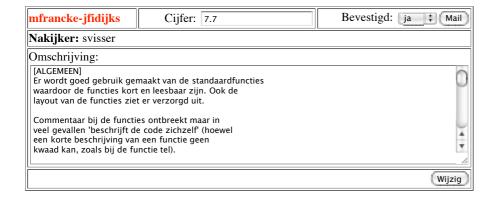


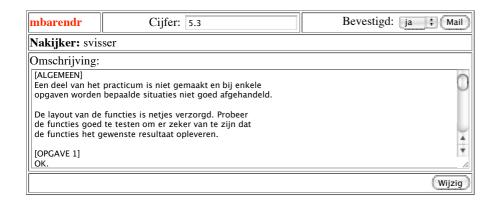




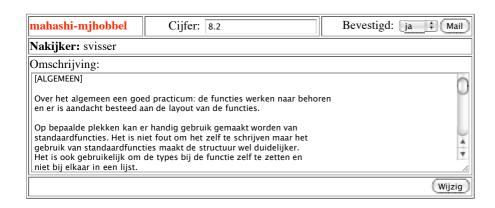












[OPGAVE 6] Prima.

sttimmer Cijfer: 10 Nakijker: bschuur Omschrijving: +De gray-codering is efficient opgelost. +De look-and-say reeksen zijn oneindig opgebouwd. +De Keith getallen zijn oneindig opgebouwd. +Je hebt goed gebruik gemaakt van lijstcomprehensies. +Je hebt goed gebruik gemaakt van high order functions. +De code-stijl is netjes. +Het commentaar is goed en effectief. Supergoed werk! Proficiat! Netjes om alle definities bovenaan in de file te zetten. Je maakt zeer goed gebruik van alle functionaliteit die in haskell zit en lost de problemen netjes en kloppend op. Nakijker: rkoot Cijfer: 10 Omschrijving: [ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentariëerd. [OPGAVE 1] Prima. [OPGAVE 2] Prima. [OPGAVE 3] Prima. [OPGAVE 4] Prima. [OPGAVE 5] Prima.

[OPGAVE 7] Prima.		
[OPGAVE 8] Prima, efficient.		
[OPGAVE 9] Prima.		
[OPGAVE 10] Prima.		
pmspek	Nakijke	er: svisser Cijfer: 10
Omschrijving: [ALGEMEEN] Weinig aan toe te voegen: een prima practicum. Er wordt goed gemaakt van standaardfuncties en accumulerende parameters of efficiëntie te verbeteren. De functies leveren ook het juiste resultaat op.	gebruik m de	
[OPGAVE 1] OK.		
[OPGAVE 2] OK.		
[OPGAVE 3] OK.		
[OPGAVE 4] OK.		
[OPGAVE 5] OK.		
[OPGAVE 6] OK.		
[OPGAVE 7] OK.		
[OPGAVE 8] OK.		
[OPGAVE 9] OK.		
[OPGAVE 10] OK.		
ewjmulde	Nakijl	ker: rkoot Cijfer: 10.0
Omschrijving: [ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentariëerd.		
[OPGAVE 1] Prima.		
[OPGAVE 2] Prima.		
[OPGAVE 3] Prima.		
[OPGAVE 4] Prima.		
[OPGAVE 5] Prima.		
[OPGAVE 6] Prima.		
[OPGAVE 7]		

[OPGAVE 8] Prima, efficient.	
[OPGAVE 9] Prima.	
[OPGAVE 10] Prima.	
lblhartm Nakijker: pdwalt	Cijfer: 1.2
Omschrijving:	
[ALGEMEEN] * nou, karig.	
+ hetgeen aanwezig was werkte wel tenminste.	
[OPGAVE 1] + werkt.	
[OPGAVE 2]	
- ontbreekt. [OPGAVE 3]	
+ werkt. [OPGAVE 4]	
- ontbreekt.	
[OPGAVE 5] - ontbreekt.	
[OPGAVE 6]	
- ontbreekt. [OPGAVE 7]	
- ontbreekt.	
[OPGAVE 8] - ontbreekt.	
[OPGAVE 9]	
- ontbreekt. [OPGAVE 10]	
- ontbreekt.	
vcmuijre Nakijker: bschuur	Cijfer: 1.5
Omschrijving: Onvolledig? Look and say was niet helemaal volgens de specificatie maar wel goed en oneindig.	
	ll.
mtduysen Nakijker: svisser	Cijfer: 2.6
mtduysen Nakijker: svisser Omschrijving:	Cijfer: 2.6
mtduysen Nakijker: svisser	Cijfer: 2.6
mtduysen Nakijker: svisser Omschrijving: [ALGEMEEN] De eerste opgaven zijn gemaakt en werken ook maar het leeuwendeel	Cijfer: 2.6
mtduysen Nakijker: svisser Omschrijving: [ALGEMEEN] De eerste opgaven zijn gemaakt en werken ook maar het leeuwendeel van het practicum ontbreekt. [OPGAVE 1]	Cijfer: 2.6
mtduysen Nakijker: svisser Omschrijving: [ALGEMEEN] De eerste opgaven zijn gemaakt en werken ook maar het leeuwendeel van het practicum ontbreekt. [OPGAVE 1] OK. [OPGAVE 2]	Cijfer: 2.6
mtduysen Omschrijving: [ALGEMEEN] De eerste opgaven zijn gemaakt en werken ook maar het leeuwendeel van het practicum ontbreekt. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3]	Cijfer: 2.6
mtduysen Nakijker: svisser Omschrijving: [ALGEMEEN] De eerste opgaven zijn gemaakt en werken ook maar het leeuwendeel van het practicum ontbreekt. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4]	Cijfer: 2.6
mtduysen Omschrijving: [ALGEMEEN] De eerste opgaven zijn gemaakt en werken ook maar het leeuwendeel van het practicum ontbreekt. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] De functie fromBase heeft het verkeerde type (Int -> [Int] -> Int) waardoor het alleen werkt voor grondtallen tot en met 10. Volgens de opgave zou het type Int -> [Char] -> Int moeten zijn, er kunnen	Cijfer: 2.6
mtduysen	Cijfer: 2.6
mtduysen	Cijfer: 2.6

[OPGAVE 10] Niet gemaakt. talles Nakijker: bschuur Cijfer: 4.0 Omschrijving: +De code-stijl is netjes. +Het commentaar is goed en effectief. Helaas, je zult nog even moeten oefenen. Vraag gerust in het practicum naar de practicumbegeleider, die kan je helpen deze problemen aan te pakken. Hoe er rekening dat je bij haskell-problemen vaak eerst op papier aan de slag kunt gaan voordat je code gaat schrijven(misschien doe je dit al, maar hoed dat in de gaten), dit is anders dan met imperatief. kjdvoors-apol Nakijker: pdwalt Cijfer: 4.1 Omschrijving: [ALGEMEEN] + code is wel bondig, MAAR gebruikt a.u.b. nooit meer namen als 'hulpfunctie101' en aanverwanten! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] - off-by-one error: *Main> G.numbers 31 ["rvhoken"] [("rvhoken",-1790769933)] zou geen resultaten moeten geven. (v==31 in base 32) [OPGAVE 8] ontbreekt. [OPGAVE 9] ontbreekt. [OPGAVE 10] + werkt. - heel erg niet oneindig. Cijfer: 4.2 lrwester Nakijker: pdwalt Omschrijving: [ALGEMEEN] - het is onhandig om zoiets als jouw zoekWaarde te doen: nu valt het wel mee omdat Haskell cached, maar anders moet je steeds door de lijst lopen. Mooier zou zijn als je bijv chr en ord zou gebruiken. + leesbare code met gebruik van standaardfuncties. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] - numbers 18 ["hyakrmln"] veroorzaakt al een stack overflow... "brute force" is hier onhandig - filter liever eerst de "woorden" die geen getallen kunnen zijn. [OPGAVE 8] - doet alsof base 2 gevraagd wordt (negeert karakters anders dan [0,1]) + niet heel inefficient [OPGAVE 9] - gaat nogal mis in het volgende voorbeeld: *Main> G.lookAndSay 111 ["9","19","1119","3119","132119", ...] [OPGAVE 10] gaat vanaf 7e element mis. Die moet namelijk 197 zijn, niet 290. + wel mooie lijst (oneindig) njvlamin Nakijker: svisser Cijfer: 4.4 Omschrijving:

[ALGEMEEN] Het eerste deel van het practicum is gemaakt en werkt grotendeels maar de latere opgaven ontbreken. [OPGAVE 1] OK. Merk op dat je met length telkens over de resterende lijst aan het lopen bent wat niet efficiënt is. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt: fromDec :: [Int] -> Int fromDec = foldl ((+) . (* 10)) 0[OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] De functie toChar is alleen gedefinieerd voor getallen en de eerste 16 getallen van het alfabet (behalve de 'a'). Dit komt omdat je voor de gegeven Int a alleen kijkt of deze groter dan 10 is en kleiner dan 26. Ook 10 hoort er bij (dat komt overeen met de letter 'a') en het zou kleiner dan 36 moeten zijn (10 cijfers + 26 letters). Als gevolg hiervan geeft de functie toBase onjuiste resultaten (vraagtekens) voor hogere grondtallen. [OPGAVE 7] De functie numbers zet wel de woorden om maar controleert niet of de woorden wel geldig zijn in het gegeven stelsel. Om het voorbeeld uit de opgave te gebruiken: "faces" is niet toegestaan in grondtal 16 want de 's' is geen geldig teken omdat alleen de getallen en 'a' t/m 'f' worden gebruikt. [OPGAVE 8] Niet gemaakt. [OPGAVE 9] Niet gemaakt. [OPGAVE 10] Niet gemaakt. jjhoozem Nakijker: pdwalt Cijfer: 4.9 Omschrijving: [ALGEMEEN] - programma code oogt wel netjes, met redelijk commentaar, maar de belangrijke functies doen het niet. jammer. [OPGAVE 1] + werkt [OPGAVE 2] - niet helemaal correct: *Getallen> toDec 1410065408 [1,1,1,4,1,0,0,6,5,4,0,8] of *Getallen> toDec 121212121212 [9,5,3,0,3,6,9,2,4]

[OPGAVE 3] + werkt [OPGAVE 4] + werkt [OPGAVE 5] + werkt [OPGAVE 6]

[OPGAVE 7]
- foutje: (off-by-one)
numbers 24 ["co"]
[("co",312)]

- heeft last van hetzelfde als toDec

moet de lege lijst teruggeven... [OPGAVE 8] - bij encoderen van grayCode wordt base als "lengte" geinterpreteerd? en decoderen geeft elke keer 2^base... [OPGAVE 9] lookAndSay 0 gaat al mis... geeft ["0", "", "" ...] OPGAVE 101 + werkt ybouma Nakijker: bschuur Cijfer: 5.0 Omschrijving: Hetgeen je gedaan hebt is kloppend en goed geschreven. Je had meer hogere-orde functies kunnen gebruiken om de code netter te krijgen. Nakijker: bschuur Cijfer: 5.0 sagieske Omschrijving: +Je hebt goed gebruik gemaakt van high order functions. +De code-stijl is netjes. -Goed dat je veel commentaar hebt geschreven maar het is teveel, probeer meer to the point te zijn en minder de code vertalen. Je numbers functie is goed maar is wat inefficiënt en zorgt zo voor stack overflows. Je lookAndSay functie was verkeerd getypeerd en werkte dus niet. rpvermeu-jcvrijho Nakijker: bschuur Cijfer: 5.0 Omschrijving: +De look-and-say reeksen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie). -Er mocht her en der wat meer commentaar bij. Je code voelt over het algemeen nog te imperatief. Bij de eerste opdrachten werkte dat ok, bij de Keith-getallen ging dat mis. Je ziet dit ook in de hoeveelhed code die je nodig hebt om problemen op te lossen. Je look-and-say reeksen zijn goed, alhoewel je erg veel conversies maakt tussen string en int, dit maakt dat er een beperkte hoeveelheid getallen gegenereerd kon worden, je had hier het type Integer inplaats van Int kunne gebruiken. In je numbers functie zit een klein foutje, alle getallen wordne geaccepteerd inclusief het grondgetal, dit is net fout. Dit kan verholpen worden door > grond te vervangen voor >= grond in de grondCheck functie. jwind Nakijker: pdwalt Cijfer: 5.2 Omschrijving: [ALGEMEEN] + programmeerstijl is netjes: korte bondige functies zonder overbodige herdefinities van standaardfuncties. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. - alleen erg langzame implementatie. [OPGAVE 8] ontbreekt. [OPGAVE 9] - lookAndSay doet helaas slechts de eerste groep in een string, namelijk bijvoorbeeld ["1","11","21","12","11"...] + wel een oneindige lijst. [OPGAVE 10] ontbreekt. mzwan-jgeeden Nakijker: svisser Cijfer: 5.2 Omschrijving: [ALGEMEEN] Probeer de functies goed te testen, ook op randgevallen, aangezien

er bij de latere opgaven verscheidene situaties niet goed worden

afgevangen.

De functies ord en chr staan in de module Char; deze kun je importeren door import Char bovenin de code te zetten. Zo hoef je ze niet letterlijk over te nemen.

[OPGAVE 1]

OK.

Merk op dat jullie met length telkens over de resterende lijst aan het lopen zijn wat niet efficiënt is. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt:

fromDec :: [Int] -> Int

fromDec = foldl ((+) . (*10)) 0

[OPGAVE 2]

OK.

[OPGAVE 3]

OK.

[OPGAVE 4]

OK.

[OPGAVE 5]

De functie fromBase houdt er geen rekening mee dat grondtallen >= 10 ook getallen kunnen bevatten. Zie ook de voorbeelden die bij deze opgave zijn gegeven. Voor grondtallen <= 9 wordt telkens een correct resultaat geleverd, bij grondtallen >= 10 werkt het alleen als er geen getallen in voorkomen.

[OPGAVE 6]

De functie toBase geeft voor het getal 10 een dubbele punt terug aangezien jullie dit geval afhandelen bij de eerste guard van deze functie. (dus xs >= 0 & x <= 10). Maar de getallen gaan van 0 t/m 9 waardoor de 10 al bij het volgende geval hoort (de 10 hoort namelijk omgezet te worden naar de letter 'a').

[OPGAVE 7]

Jullie gebruiken de functie fromBase om te kijken of een woord wel geldige tekens bevat. Deze functie zal wel voor ongeldige letters de waarde 0 opleveren (bijvoorbeeld fromBase 16 "s") maar voor hele woorden die gedeeltelijk ook goede tekens bevatten, wordt er een waarde ongelijk aan 0 opgeleverd (bijvoorbeeld fromBase 16 "faces" geeft 1027296). Op deze manier worden ongeldige woorden toch toegestaan. Het was beter geweest om expliciet te controlen of woorden geldig waren.

[OPGAVE 8]

Niet gemaakt.

[OPGAVE 9]

Doordat jullie een Int gebruiken om deelresultaten in op te slaan, houdt de functie er al vrij gauw mee op aangezien de grotere getallen niet meer in een Int passen. Het was beter geweest om lijsten te gebruiken aangezien die wel kunnen doorgroeien.

[OPGAVE 10]

De functie keithGetal werkt niet voor getallen met drie of meer cijfers aangezien jullie alleen de eerste twee cijfers gebruiken. Zo begint de reeks van 197 met:

[1,9,10,19,29,48,77,125,202,327

terwijl deze zou moeten beginnen met [1,9,7,...].

Ook ontbrak de functie keithGetallen terwijl die niet zo moeilijk meer is als je al de functie hebt om de reeks te produceren. Bijvoorbeeld:

keithGetallen :: [Int]

keithGetallen = filter isKeithGetal [10..]

isKeithGetal :: Int -> Bool

isKeithGetal n = elem n (takeWhile (<= n) (keithGetal n))

De functie elem kijkt of het gegeven element in een lijst voorkomt. Het kan efficiënter aangezien we al weten dat, als n voorkomt, deze zich aan het eind zal bevinden maar het gaat even om het idee.

jcgoosen Nakijker: pdwalt Cijfer: 5.3

Omschrijving:

[ALGEMEEN]

- * layout is niet heel netjes (inline ifjes enz.)
- grayCode werkte niet zoals aangeleverd, patch was nodig.
- geen types aangegeven

[OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4] + werkt [OPGAVE 5] + werkt [OPGAVE 6] + werkt [OPGAVE 7] - failed alle tests: probeer numbers 31 ["judjzzcjrlagrq","riv"] eens (z mag er niet in voorkomen). [OPGAVE 8] - fromGray ontbreekt [OPGAVE 9] ontbreekt [OPGAVE 10] + werkt + oneindige lijst Cijfer: 5.3 mbarendr Nakijker: svisser Omschrijving: [ALGEMEEN] Een deel van het practicum is niet gemaakt en bij enkele opgaven worden bepaalde situaties niet goed afgehandeld. De layout van de functies is netjes verzorgd. Probeer de functies goed te testen om er zeker van te zijn dat de functies het gewenste resultaat opleveren. [OPGAVE 1] OK. Merk op dat je met length telkens over de resterende lijst aan het lopen bent wat niet efficiënt is. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt: fromDec :: [Int] -> Int |fromDec = foldl ((+) . (* 10)) 0[OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] De functie fromBase geeft een nul als het hele woord bestaat uit ongeldige tekens; als het woord gedeeltelijk ook toegestane tekens bevat, dan zal de functie numbers wel het woord opnemen in het eindresultaat. Om het voorbeeld uit de opgave te gebruiken: numbers 16 ["sss"] geeft [] (correct) numbers 16 ["faces"] geeft [("faces",1027296)] (incorrect) De 's' is immers niet toegestaan bij grondtal 16 (daar zijn alleen de getallen en 'a' t/m 'f' toegestaan).

[OPGAVE 8] Niet gemaakt. [OPGAVE 9] Niet gemaakt.

[OPGAVE 10]

De functie keithGetallen levert geen resultaat op omdat de functie isKeithGetal niet eindigt voor niet-Keithgetallen. Dit komt omdat de functie genKeithGetallen niets oplevert: de functie blijft zichzelf aanroepen met een steeds groter argument.

Je aanpak werkt verder wel, als de functie genKeithGetallen zo wordt geschreven:

genKeithGetallen :: [Int]->Int->[Int]

genKeithGetallen 1 size = csum : genKeithGetallen (1 ++ [csum]) size

where csum = sum (take size (reverse 1))

Overigens, de functie vanaf is hetzelfde als

iterate (+1) 1. Het opschrijven van deze lijst kan ook

zo: [1..], dit produceert ook de oneindige lijst vanaf 1.

jcgsmits Nakijker: pdwalt Cijfer: 5.4

Omschrijving:

[ALGEMEEN]

- + code ziet er redelijk netjes uit,
- maar veel dingen werken net niet soepel.

[OPGAVE 1]

- + werkt
- [OPGAVE 2]
- kapot: probeer eens toDec 1000000000

[OPGAVE 3]

- + werkt
- [OPGAVE 4]
- werkt meestal, behalve af en toe een divide-by-zero:
- > toBin 1073741824

[OPGAVE 5]

- + werkt
- [OPGAVE 6]
- termineert niet altijd, maar meestal wel correct.

[OPGAVE 7]

- + werkt
- [OPGAVE 8]
- ontbreekt

[OPGAVE 9]

- + oneindige lijst
- incorrect: lookAndSay 0 geeft

["0","10","1110","3110","211310","789474889" ...] maar moet zijn

 $["0","10","1110","3110","132110","1113122110","311311222110", \ldots]$

[OPGAVE 10]

- + oneindige lijst geconstrueerd
- + werkt

tleroi Nakijker: bschuur Cijfer: 5.5

Omschrijving

- +De look-and-say reeksen zijn oneindig opgebouwd, maar lopen tegen de beperkingen van de Int aan.
- -Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie).
- -Je code is soms wat rommelig, vooral de naamgeving is onduidelijk(toBase2, lookAndSay3).
- -Commentaar is volledig afwezig.

De opbouw van je functies doet vreemd aan. Bij elke functie heb je een functie2 geschreven die aangeroepn wordt. Deze hebben wel nut maar zou je het problem niet zo kunnen omschrijven dat slechts 1 functie nodig is? Verder zorgt dit ervoor dat je niet alle gevallen afdekt, hier loop je o.a. tegen aan in je look-and-say functie. Je numbers functie maakt helaas geen selectie.

rhouweli Nakijker: bschuur Cijfer: 5.5

Omschrijving:

- +Je hebt goed gebruik gemaakt van high order functions.
- +Het commentaar is goed en effectief.
- +De code-stijl is netjes.

In de numbers functie maak je een kleine fout door alle getallen –inclusief- het grondtal goed te rekenen(dus bij grondtal 35 wordt "z" goedgekeurd). Dit is heel snel te verhelpen door in je aannemelijk functie < x inplaats van <= x te gebruiken.

kfaro Nakijker: pdwalt Cijfer: 5.5

Omschrijving:

[ALGEMEEN]

- + veel commentaar, leesbare code,
- + veel gebruik van standaardfuncties

```
[OPGAVE 1]
 + werkt.
[OPGAVE 2]
+ werkt
[OPGAVE 3]
+ werkt
[OPGAVE 4]
+ werkt
[OPGAVE 5]
+ werkt
[OPGAVE 6]
+ werkt
[OPGAVE 7]
- gek hoor, het volgende:
*Main> G.numbers 33 ["hkanwprh"]
 *Main> G.numbers 33 ["hlanwprh"]
[("hlanwprh",434889388)]
een makkelijkere manier is om woorden weg te strepen met cijfers buiten het
toegestane alfabet.
[OPGAVE 8]
 - ontbreekt.
(er werd inderdaad n-ary Gray Coding bedoeld)
[OPGAVE 9]
 - verkeerd type. uitvoer was echter wel correct, dus niet zwaar aangerekend.
 - helaas ook niet werkend als lookAndSay 0.
+ oneindige lijst.
[OPGAVE 10]

    ontbreekt. bedankt voor de eerlijkheid.

jleersum-anieuwla
                                                                                                  Nakijker: pdwalt
                                                                                                                                           Cijfer: 5.5
Omschrijving:
[ALGEMEEN]
+ leesbare code met commentaar. goed gebruik van standaardfuncties.
[OPGAVE 1]
+werkt
[OPGAVE 2]
+werkt
[OPGAVE 3]
+werkt
[OPGAVE 4]
+werkt
[OPGAVE 5]
+werkt
[OPGAVE 6]
+werkt
[OPGAVE 7]
 - ook niet helemaal kosher:
*Getallen> numbers 30 ["akejkda"]
terwijl k < 30
[OPGAVE 8]
 ontbreekt
[OPGAVE 9]
 - zo tegen element 8 gaat lookAndSay 0 de mist in.
 *Getallen> take 10 $ lookAndSay 0
\lceil \lceil 0 \rceil, \lceil 10 \rceil, \lceil 1110 \rceil, \lceil 3110 \rceil, \lceil 132110 \rceil, \lceil 1113122110 \rceil, \lceil 311311222110 \rceil, \lceil 12101713151716171918 \rceil, \lceil 111814101711171926 \rceil, \lceil 211613182932 \rceil
["0","10","110","3110","3110","132110","1113122110","311311222110","13211321322110","111312211312113222110","31131122211311123113322110"]
[OPGAVE 10]
 - keithGetallen begint al fout, bij 17 als eerste element.
 + wel oneindige lijst
jgageldo
                                                                                                  Nakijker: pdwalt
                                                                                                                                           Cijfer: 5.6
Omschrijving:
[ALGEMEEN]
+ programmacode ziet er verzorgd uit, inclusief commentaar
[OPGAVE 1]
+ werkt
[OPGAVE 2]
+ werkt
[OPGAVE 3]
+ werkt
[OPGAVE 4]
+ werkt
[OPGAVE 5]
```

```
+ werkt
[OPGAVE 6]
+ werkt
[OPGAVE 7]
- werkt niet met base = 36,
+ verder wel
[OPGAVE 8]
ontbreekt
[OPGAVE 9]
- kapot, geeft:
lookAndSay 0
["0","10","1110","3110","132110","1113122110","2073576798","1840717966","1163899222","1159785428","-348524702","***
Exception: Prelude.undefined
[OPGAVE 10]
+ werkt
+ mooie oneindige lijst
fcbijlsm
                                                                                                                               Cijfer: 5.7
                                                                                          Nakijker: rkoot
Omschrijving:
[ALGEMEEN]
Code is slorig.
Weing gebruik gemaak van standaard- en hulpfuncties.
Code is goed becommentariëerd.
Dit had efficienter geimplementeerd kunnen worden zonder gebruik te maken van machtsverheffen.
[OPGAVE 2]
Prima.
[OPGAVE 3]
Prima.
[OPGAVE 4]
Prima.
[OPGAVE 5]
Prima.
[OPGAVE 6]
Prima.
[OPGAVE 7]
Werkt niet altijd correct, bijvoorbeeld voor het geval numbers 32 ["wqpm","n"].
[OPGAVE 8]
[OPGAVE 9]
[OPGAVE 10]
Prima.
jhberg
                                                                                         Nakijker: pdwalt
                                                                                                                                 Cijfer: 6
Omschrijving:
[ALGEMEEN]
- types vergeten boven lookAndSay
+ geen lelijke code, maar let wel op dingen als types expliciet vermelden.
+ veel commentaar
[OPGAVE 1]
+werkt
[OPGAVE 2]
+werkt
[OPGAVE 3]
+werkt
[OPGAVE 4]
+werkt
[OPGAVE 5]
+werkt
[OPGAVE 6]
+werkt
[OPGAVE 7]
- ontbreekt
[OPGAVE 8]
- helaas hebben Graycodes leading "0", maar verder in principe ok.
+ redelijk efficient gedaan
```

[OPGAVE 9]

- + werkt
- + oneindige lijst

[OPGAVE 10]

- verkeerd, zelfs de eerste paar elementen kloppen niet met de beschrijving.
- + oneindige lijst

pdstaats-cjplatte Nakijker: svisser Cijfer: 6.0

Omschrijving:

[ALGEMEEN]

Bij de meeste opgaven hebben jullie een werkende aanpak bedacht. Er wordt niet heel veel gebruik gemaakt van standaardfuncties - het zelf schrijven mag natuurlijk maar doorgaans wordt de structuur beter zichtbaar als men een standaardfunctie gebruikt, zoals map en filter.

Het commentaar is bij tijden een beschrijving van de code maar dan in natuurlijke taal. Commentaar in broncode is o.a. bedoeld om de achterliggende ideeën toe te lichten aangezien men mag aannemen dat ander de Haskell-code kan lezen. Zo kan het zinvol zijn om kort uit te leggen hoe men de Gray-codering construeert, de daadwerkelijke implementatie kan men dan eronder lezen.

Wat betreft het ontbreken van de Gray-codering; misschien dat jullie aan de begeleiders en/of studentassistenten om uitleg hadden kunnen vragen over een aanpak?

[OPGAVE 1]

OK.

[OPGAVE 2]

OK.

[OPGAVE 3]

OK.

[OPGAVE 4]

OK.

[OPGAVE 5]

OK.

[OPGAVE 6]

Jullie hergebruiken de functie convertInt maar deze is slechts gedefinieerd tot base 16. Voor grondtallen > 16 gaat de functie toBase dus mis aangezien het niet meer het teken kan bepalen voor die hogere Int-waarden.

Voor het omzetten van een Char naar een Int en andersom bestaan de functies ord en chr (in de module Char) - de functies fromEnum en toEnum kunnen dit ook maar ord en chr geven in dit verband beter weer wat je bedoelt.

[OPGAVE 7]

OK.

Bij deze functie kan men bijvoorbeeld de functies filter, om de geldige woorden uit de lijst te halen, en map, om een woord om te zetten naar een tupel, gebruiken.

[OPGAVE 8]

Deze opgave is niet gemaakt.

[OPGAVE 9]

Het gebruik van een Int om het resultaat in op te slaan, zorgt er snel voor dat de waarden niet meer te bewaren zijn (het past simpelweg niet). Het was daarom beter geweest om lijsten te gebruiken voor het 'beschrijfproces' aangezien die wel kunnen doorgroeien. Deze kunnen vervolgens omgezet worden naar de Strings die in het eindresultaat terechtkomen. Verder werkt de gebruikte aanpak wel.

[OPGAVE 10]

De functie werkt wel maar hij levert pas resultaten op als ie de gehele lijst bestudeerd heeft. Om dit te voorkomen zou je niet de hele lijst moeten beschouwen maar per getal kijken of het een Keithgetal is, dus een functie:

isKeithGetal :: Int -> Bool isKeithGetal x = ...

Vervolgens kun je met filter de Keithgetallen er uit halen:

```
keithGetallen :: [Int]
keithGetallen = filter isKeithGetal [10..]
jmlinsse
                                                                                             Nakijker: pdwalt
                                                                                                                                    Cijfer: 6.3
Omschrijving:
[ALGEMEEN]
+ nette code, erg veel commentaar.
+ we worden blij van standaardfuncties.
[OPGAVE 1]
+ werkt
[OPGAVE 2]
+ werkt
[OPGAVE 3]
+ werkt
[OPGAVE 4]
+ werkt
[OPGAVE 5]
+ werkt
[OPGAVE 6]
+ werkt
[OPGAVE 7]
- off-by-one error in numbers, wat betreft de base: namelijk
*Main> G.numbers 25 ["hdpl","hux","rbv"]
[("hdpl",274396)]
terwijl p == 25, in base 26!
[OPGAVE 8]
- ontbreekt
[OPGAVE 9]
- jammer, niet helemaal het idee, er werd bedoeld dat een (decimaal) getal werd
 'voorgelezen", waarna de reeks die daaruit kwam ook moest worden voorgelezen,
enz. dus 36 -> ["36","1316" ..]. Jouw versie klopt namelijk soort van, maar
dan met base > 10. Toch wat puntjes.
+ goede oneindige lijst.
[OPGAVE 10]
+ werkt
idfeddem
                                                                                                                                    Cijfer: 6.3
                                                                                            Nakijker: pdwalt
Omschrijving:
[ALGEMEEN]
+ goed gebruik van commentaar, plus mooie layout
[OPGAVE 1]
+ werkt
[OPGAVE 2]
+ werkt
[OPGAVE 3]
+ werkt
[OPGAVE 4]
+ werkt
[OPGAVE 5]
+ werkt
[OPGAVE 6]
+ werkt
[OPGAVE 7]
- numbers 35 ["kufitbctzerl"] geeft een antwoord, terwijl "z" niet mag; komt door
off-by-one error.
+ klopt verder wel
[OPGAVE 8]
incorrect, heen en weer gaat kapot:
*Getallen> (snd $ grayCode 4) 16
"133"
*Getallen> (fst $ grayCode 4) "133"
opeenvolgende getallen moeten 1 plaats verschillen:
*Getallen> (snd$grayCode 2) 110012
"10011011011010111"
*Getallen> (snd$grayCode 2) 110011
"10011011011010010"
+ verder wel redelijk efficient geimplementeerd.
[OPGAVE 9]
+ oneindige lijst
- lookAndSay 10 geeft "a" als eerste resultaat, dit moet natuurlijk "10" zijn...
Weer off-by-one.
[OPGAVE 10]
+ keithGetallen werkt op zich wel, maar heel langzaam en inefficient.
```

Omschrijving:		Nakijker: pdwalt	Cijfer: 6.3			
[ALGEMEEN] + code maakt veel gebruik van standaardfuncties en i	is goed becommentarieerd					
[OPGAVE 1]	is goed becommendational	•				
+ werkt.						
[OPGAVE 2] + werkt.						
[OPGAVE 3]						
+ werkt.						
[OPGAVE 4] + werkt.						
[OPGAVE 5]						
+ werkt. [OPGAVE 6]						
+ werkt.						
[OPGAVE 7]						
- hoe is dit correct? *Main> G.numbers 24 ["jz"]						
makkelijker is waarschijnlijk gewoon wegstrepen wat	t niet mag volgens base-al	fabet.				
[OPGAVE 8] - niet zijn eigen inverse!						
- ook erg inefficiente implementatie						
[OPGAVE 9] - jammer, dit:						
*Main> G.lookAndSay 9						
[[""]						
in jullie commentaar staat dat het anders niet klopt, no ["19","1119"] maken	ou je kan met 9 gewoon					
+ verder werkt het wel.						
[OPGAVE 10]						
+ werkt. + mooie lijst-oplossing.						
rslagmol		Nakijker: bschuur	Cijfer: 6.5			
Omschrijving:			2-5			
+Je hebt goed gebruik gemaakt van high order functions. Je had hier nog meer gebruik van kunnen maken bij de eerste opgaven. -Je code is soms wat rommelig, bijvoorbeeld een functie 'rekenen a z y x' is compleet onduidelijk wat het doet. -Er mocht her en der wat meer commentaar bij. Ik mis het commentaar bij juist de moeilijke gedeeltes. Bij de toBin en toBase functies verslikt hij zich als 0 wordt ingegeven, dan onstaan negatieve machten. Dit komt omdat je de binnengekomen waarde verlaagd naar de hulpfuncties doorgeeft en deze dan niet wordt afgevangen door een guard. Je numbers functie verslikt zich in het randgeval waarbij er een letter voorkomt die in waarde gelijk is aan het grondtal, deze wordt toegestaan maar zou moeten wordne afgewezen.						
er een letter voorkomt die in waarde gelijk is aan het		or een guard. Je numbers functie verslikt zich in he	nengekomen waarde et randgeval waarbij			
		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding		or een guard. Je numbers functie verslikt zich in he	nengekomen waarde et randgeval waarbij Cijfer: 6.7			
jeceding Omschrijving:		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1]		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt.		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt.		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3]		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt.		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt.		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5]		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5]		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt.		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5]		or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8]	grondtal, deze wordt toeg	or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8] + last van het lege String in plaats van "0" probleem.	grondtal, deze wordt toeg	or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen.	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8] + last van het lege String in plaats van "0" probleem wel goed over nagedacht. [OPGAVE 9]	grondtal, deze wordt toeg	or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen. Nakijker: pdwalt	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8] + last van het lege String in plaats van "0" probleem wel goed over nagedacht. [OPGAVE 9] - kop klopt niet helemaal -> lookAndSay 0 = ["" , "10"	grondtal, deze wordt toeg	or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen. Nakijker: pdwalt	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8] + last van het lege String in plaats van "0" probleem wel goed over nagedacht. [OPGAVE 9] - kop klopt niet helemaal -> lookAndSay 0 = ["" , "10 maar verder bijna juist	grondtal, deze wordt toeg	or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen. Nakijker: pdwalt	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8] + last van het lege String in plaats van "0" probleem wel goed over nagedacht. [OPGAVE 9] - kop klopt niet helemaal -> lookAndSay 0 = ["" , "10 maar verder bijna juist + oneindige lijst [OPGAVE 10]	verder bijna ok. O"] moet met "0" beginn	or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen. Nakijker: pdwalt	et randgeval waarbij			
jeceding Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8] + last van het lege String in plaats van "0" probleem wel goed over nagedacht. [OPGAVE 9] - kop klopt niet helemaal -> lookAndSay 0 = ["" , "10 maar verder bijna juist + oneindige lijst	verder bijna ok. O"] moet met "0" beginn	or een guard. Je numbers functie verslikt zich in heestaan maar zou moeten wordne afgewezen. Nakijker: pdwalt	et randgeval waarbij			

lwgraaff	Nakijker: pdwalt	Cijfer: 6.7
Omschrijving:		
[ALGEMEEN] + code handig afgebakend in secties; standaardfuncties worden	gebruikt (foldl, yay).	
+ commentaar. [OPGAVE 1]		
+ werkt.		
[OPGAVE 2] + werkt.		
[OPGAVE 3]		
+ werkt. [OPGAVE 4]		
+ werkt.		
[OPGAVE 5] + werkt.		
[OPGAVE 6]		
- kapot: *Main> G.toBase 30 10		
"k" moet natuurlijk "a" opleveren.		
[OPGAVE 7]		
- oei, numbers vindt dit base-10: *Main> G.numbers 10 ["jed"]		
[("jed",2053)]		
[OPGAVE 8]		
+ het encoderen lijkt goed te gaan,		
- maar het decoderen is bijna de identiteitsfunctie. + niet heel inefficient		
[OPGAVE 9] + werkt.		
[OPGAVE 10]		
+ werkt. + mooie oneindige lijst.		
dwinkel	Nakijker: rkoot	Cijfer: 6.8
Omschrijving:	J	
[ALGEMEEN] Code is duidelijk leesbaar.		
Handig gebruik gemaak van standaard- en hulpfuncties.		
Code is voldoende becommentariëerd.		
[OPGAVE 1]	bruik ta makan yan machtayarhaffan	
Dit had efficienter geimplementeerd kunnen worden zonder ge	ordik të maken van maentsverhenen.	
[OPGAVE 2] Prima.		
[OPGAVE 3] Prima.		
[OPGAVE 4] Prima.		
[OPGAVE 5] Prima.		
[OPGAVE 6]		
Prima.		
[OPGAVE 7]		
Prima.		
[OPGAVE 8]		
-		
[OPGAVE 9]		
Werkt, maar is te inefficient om meer dan een handvol look-an	d-say te generenen.	
[OPGAVE 10]		
Werkt, maar is te inefficient om meer dan een handvol Keith-g		O (0
bgreeven-jsteenbe Omschrijving:	Nakijker: rkoot	Cijfer: 6.9
[ALGEMEEN]		
Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties.		

[OPGAVE 1]		
Prima.		
[OPGAVE 2] Prima.		
[OPGAVE 3] Prima.		
[OPGAVE 4] Jullie antwoord wordt achterstevoren opgeleverd.		
[OPGAVE 5] Prima.		
[OPGAVE 6] Prima.		
[OPGAVE 7] Prima.		
[OPGAVE 8]		
[OPGAVE 9] Werkt niet voor startwaardes groter dan 9.		
[OPGAVE 10] Prima.		
taveld	Nakijker: bschuur	Cijfer: 7.0
Alhoewel al je to* functies lijken te kloppen kunnen ze niet go zit(waarschijnlijk beperkingen van int). Je getCharacter functie die gelijk zijn aan het grondtal, dit moeten echter alle letters zi	e in toBase bevat een onvolledig patroon. Je numbers f	
mmontvai-beck	ngh die eertje minder zijn dan het grondtal. Nakijker: svisser	Cijfer: 7.0
	Nakijker: svisser	
mmontvai-beck Omschrijving: [ALGEMEEN] Over het algemeen een goed practicum; de functies zijn kort door het gebruik van standaardfuncties en hulpfuncties. De Gray-opgave ontbreekt en er zijn enkele situaties waarbij een verkeerd resultaat wordt teruggegeven maar verder werkt len be code maakt wel veel gebruik van if then else constructies	Nakijker: svisser	
mmontvai-beck Omschrijving: [ALGEMEEN] Over het algemeen een goed practicum; de functies zijn kort door het gebruik van standaardfuncties en hulpfuncties. De Gray-opgave ontbreekt en er zijn enkele situaties waarbij een verkeerd resultaat wordt teruggegeven maar verder werkt l De code maakt wel veel gebruik van if then else constructies terwijl het soms ook 'functioneler' kan. [OPGAVE 1]	Nakijker: svisser	
mmontvai-beck Omschrijving: [ALGEMEEN] Over het algemeen een goed practicum; de functies zijn kort door het gebruik van standaardfuncties en hulpfuncties. De Gray-opgave ontbreekt en er zijn enkele situaties waarbij een verkeerd resultaat wordt teruggegeven maar verder werkt l De code maakt wel veel gebruik van if then else constructies terwijl het soms ook 'functioneler' kan. [OPGAVE 1] OK.	Nakijker: svisser	
omschrijving: [ALGEMEEN] Over het algemeen een goed practicum; de functies zijn kort door het gebruik van standaardfuncties en hulpfuncties. De Gray-opgave ontbreekt en er zijn enkele situaties waarbij een verkeerd resultaat wordt teruggegeven maar verder werkt le De code maakt wel veel gebruik van if then else constructies terwijl het soms ook 'functioneler' kan. [OPGAVE 1] OK. [OPGAVE 2] OK.	Nakijker: svisser	
mmontvai-beck Omschrijving: [ALGEMEEN] Over het algemeen een goed practicum; de functies zijn kort door het gebruik van standaardfuncties en hulpfuncties. De Gray-opgave ontbreekt en er zijn enkele situaties waarbij een verkeerd resultaat wordt teruggegeven maar verder werkt l De code maakt wel veel gebruik van if then else constructies terwijl het soms ook 'functioneler' kan. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK.	Nakijker: svisser	
omschrijving: [ALGEMEEN] Over het algemeen een goed practicum; de functies zijn kort door het gebruik van standaardfuncties en hulpfuncties. De Gray-opgave ontbreekt en er zijn enkele situaties waarbij een verkeerd resultaat wordt teruggegeven maar verder werkt li De code maakt wel veel gebruik van if then else constructies terwijl het soms ook 'functioneler' kan. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK.	Nakijker: svisser	

Voor lagere grondtallen worden de getallen ook toegestaan, dus bijvoorbeeld bij grondtal 5 mogen ook 5, 6, 7, 8 en 9 gebruikt worden.

[OPGAVE 8]

Deze opgave is niet gemaakt. Misschien kunnen jullie om uitleg en/of tips vragen bij de begeleiders of studentassistenten bij het practicum?

[OPGAVE 9]

De functie lookAndSay geeft al vrij gauw verkeerde getallen aangezien jullie de deelresultaten opslaan in een Int. De grotere getallen passen hier niet meer in waardoor er verkeerde waarden worden opgeleverd. Het was beter geweest om met lijsten te blijven werken aangezien die wel kunnen doorgroeien.

De functie lookAndSay werkt ook niet voor 0 als begingetal.

[OPGAVE 10]

OK.

Bij meerdere if then else constructies kun je beter guards gebruiker, dat is overzichtelijker en geeft ook mooier Haskell.

Dit gebeurt ook bij de functie convertToInt eerder.

hjkuijk Nakijker: rkoot Cijfer: 7.1

Omschrijving:

[ALGEMEEN]

Code is duidelijk leesbaar.

Weinig gebruik gemaak van standaardfuncties.

Code is onvoldoende becommentariëerd.

[OPGAVE 1]

Dit had efficienter gedaan kunnen worden zonder gebruik te maken van machtsverheffen.

[OPGAVE 2]

Prima.

[OPGAVE 3]

Prima.

[OPGAVE 4]

Prima.

[OPGAVE 5]

Prima.

[OPGAVE 6]

Prima.

[OPGAVE 7]

Prima.

[OPGAVE 8]

Prima.

[OPGAVE 9]

Werkt niet goed indien startwaarde al groter dan 9.

[OPGAVE 10]

Prima.

ddtoniss-dsgroote Nakijker: rkoot Cijfer: 7.2

Omschrijving:

[ALGEMEEN]

Code is redelijk leesbaar.

Handig gebruik gemaak van standaard- en hulpfuncties.

Code is goed becommentariëerd.

[OPGAVE 1]

Prima.

[OPGAVE 2]

Prima.

[OPGAVE 3]

fromBin: Een oplossing zonder gebruik te maken van machtsv	verheffen was efficiënter geweest.	
[OPGAVE 4] Prima		
[OPGAVE 5] parsePowers: Hier hadden jullie gebruik kunnen maken van de	e standaardfunctie "ord".	
[OPGAVE 6] Prima.		
[OPGAVE 7] De functie is incorrect: * Hij had wel degelijk moeten werken op getallen met een gro * Er zit een off-by-one error in jullie implementatie (numbers	ondgetal kleiner dan 10, ook al produceert dit altijd een lege lijst. 16 ["f", "g", "h"] levert [("f",15),("g",16)] op in plaats van [("f",15)]).	
[OPGAVE 8] Jullie implentatie komt door geen enkele van onze testcases he	een en is niet erg efficiënt geïmplementeerd.	
[OPGAVE 9] Prima.		
[OPGAVE 10] Prima.		
dpboot	Nakijker: rkoot	Cijfer: 7.3
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Weinig gebruik gemaak van standaardfuncties. Code is goed becommentariëerd. [OPGAVE 1] Dit had efficienter geimplementeerd kunnen worden zonder ge [OPGAVE 2] Prima. [OPGAVE 3] Prima. [OPGAVE 4] Prima. [OPGAVE 5] Prima. [OPGAVE 5] Prima. [OPGAVE 7]	ebruik te maken van machtsverheffen.	
Prima. [OPGAVE 8] - [OPGAVE 9] Werkt niet voor startwaarden groter dan 9.		
[OPGAVE 10] Prima.		
mtibboel-sabitter	Nakijker: svisser	Cijfer: 7.3
Omschrijving: [ALGEMEEN] De code bestaat uit korte functies doordat er hulpfuncties word gebruikt waar nodig. Over het algemeen een goed practicum ner zijn enkele dingen die niet goed worden afgehandeld (bijvoorbeeld door de gekozen aanpak, bij opgave 9, of door randgevallen, zoals bij opgave 2). [OPGAVE 1] OK. [OPGAVE 2]		
De functie toDec geeft een negatief begingetal voor getallen		

die beginnen met 10 (dus 10, 1000 of 10123). Het otherwise geval wordt dan uitgevoerd waardoor het getal negatief wordt. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] Het opsporen van ongeldige woorden had gedaan kunnen worden door zelf een functie te schrijven die controleert of een woord geldig is (i.p.v. van fromBase gebruik te maken). Vervolgens kun je met een filter de geldige woorden eruit filteren: numbers base words = map maakTupel (filter isGeldig words) where maakTupel $x = (x, \hat{f}romBase \hat{b}ase x)$ isGeldig w = ... Een manier om de functie isGeldig te schrijven kan door een alphabet te definiëren: alphabet :: [Char] alphabet = ['0'..'9'] ++ ['a'..'z']De geldige tekens in een grondtal kunnen nu worden geschreven als: alphBase :: Int -> [Char] alphBase base = take base alphabet De functie isGeldig zou nu voor elk teken in het meegegeven woord moeten kijken of het in (alphBase base) voorkomt. Het kan efficiënter dan dit maar het gaat even om het idee. Een efficiëntere manier is om per teken in het meegegeven woord te kijken of de Int-waarde van het character in een geldige range ligt (die bij dat grondtal hoort). [OPGAVE 8] De functie toGray en fromGray komen niet altijd overeen; het zou zo moeten zijn dat het resultaat van toGray weer ingelezen kan worden met fromGray om dan het oorspronkelijke getal op te leveren. Dit gebeurt niet altijd, zie bijvoorbeeld: toGray 7 49 (geeft "166") en fromGray 7 "166" (geeft 55, zou 49 moeten zijn). [OPGAVE 9] De functie geeft na enkele elementen verkeerde resultaten aangezien de deelresultaten in een Int worden opgeslagen. Het was beter om met lijsten te werken aangezien die wel kunnen doorgroeien (een Int is immers maar beperkt qua opslag). take 10 (lookAndSay 1) geeft bijvoorbeeld (zie de laatste hiervan): ["1","11","21","1211","111221","312211","13112221","1113213211" ,"31131211131221","111228111619181113"] Als gevolg hiervan levert de functie ook geen oneindige lijst op. [OPGAVE 10] OK.

x 1 1.1 .

Merk op dat keithGetallen ook geschreven had kunnen worden als filter keithGetal [11..].

jhorn Nakijker: pdwalt Cijfer: 7.4

Omschrijving:

[ALGEMEEN]

+ nette becommentarieerde code, soms een beetje spaghetti-achtig

[OPGAVE 1]

+ werkt

[OPGAVE 2]

+ werkt

[OPGAVE 3] + werkt [OPGAVE 4] + werkt [OPGAVE 5] + werkt [OPGAVE 6] + werkt [OPGAVE 7]	
+ werkt [OPGAVE 8] - ontbreekt [OPGAVE 9] - gaat heel langzaam bij geringe waarden, neem bv. lookAndSaautomatisch testen faalt)	ay 789. (waardoor
+ klopt wel voor lagere beginwaarden. [OPGAVE 10]	
+ werkt wokatz	Nakijker: bschuur Cijfer: 7.5
	basisgeval 0 niet afgevangen wat het programma in een oneindige loop gooit. Verder de waarde als het grondtal worden nog steeds als kloppend gezien(dit moet 1 minder
zijn).	
sjveldhu	Nakijker: bschuur Cijfer: 7.5
+De look-and-say reeksen zijn oneindig opgebouwd, maar gaa +De Keith getallen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p. +De code-stijl is netjes. +Het commentaar is goed en effectief.	t op een gegeven moment de fout in (Integer inplaats van Int gebruiken?). v. recursie, lijstcomprehensie). Zeker bij de eerste opgaven.
rvesten-rsalphen	Nakijker: bschuur Cijfer: 7.5
Omschrijving: +De look-and-say reeksen zijn oneindig opgebouwd. +De Keith getallen zijn oneindig opgebouwdJe had meer high order functions kunnen gebruiken (folds i.p. +De code-stijl is netjesCommentaar is volledig afwezig.	v. recursie, lijstcomprehensie).
Nette code-stijl, jammer dat de graycode niet gelukt is. De lool iets met de standaard-bibliotheken).	k-and-say reeksen liepen bij mij vast maar zien er qua opbouw goed uit (waarschijnlijk
ldspbroe-jfklein	Nakijker: pdwalt Cijfer: 7.6
Omschrijving: [ALGEMEEN] + programmeerstijl is redelijk netjes, leesbare code. + commentaar. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 6] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + to start the start that the st	

meegerekend). + werkt wel ok gegeven dat feit. [OPGAVE 9] + werkt. [OPGAVE 10] + werkt. jmulder Nakijker: pdwalt Cijfer: 7.6 Omschrijving: [ALGEMEEN] - geen noemenswaardig commentaar, dat is jammer. + wel nette (leesbare) code met gebruik van standaardfuncties. [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4] + werkt [OPGAVE 5] + werkt [OPGAVE 6] + werkt [OPGAVE 7] + werkt [OPGAVE 8] - alleen "encodeerfunctie" meegeleverd, maar die werkt ook niet: *Main> map (grayChar 2) [0..6] ["0","","1","0","010","110","110"] dus. [OPGAVE 9] + werkt + oneindige lijst [OPGAVE 10] + werkt + oneindige lijst Cijfer: 7.6 bdoren-mjspoor Nakijker: rkoot Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Voldoende commentaar. [OPGAVE 1] Prima. [OPGAVE 2] Prima. [OPGAVE 3] Prima. [OPGAVE 4] Prima. [OPGAVE 5] Prima. [OPGAVE 6] Werkt niet voor het randgeval 0. [OPGAVE 7] Prima. [OPGAVE 8] [OPGAVE 9] Werkt niet voor het randgeval 0. [OPGAVE 10] Prima. gcpzunde Cijfer: 7.6 Nakijker: rkoot Omschrijving:

[ALGEMEEN] Code is duidelijk leesbaar. Weinig gebruik gemaak van standaardfuncties. Code is goed becommentariëerd. [OPGAVE 1] Dit had efficienter geimplementeerd kunnen worden zonder gebruik te maken van machtsverheffen. [OPGAVE 2] Prima. [OPGAVE 3] Prima. [OPGAVE 4] Prima. [OPGAVE 5] Prima. [OPGAVE 6] Prima. [OPGAVE 7] Prima. [OPGAVE 8] [OPGAVE 9] Werkt niet voor het randgeval 0. [OPGAVE 10] Prima. Cijfer: 7.6 ravries-ccmrooij Nakijker: svisser Omschrijving: [ALGEMEEN] De functies werken naar behoren. De functies zijn wel erg breed opgeschreven: probeer gemeenschappelijke dingen in een where te zetten zodat de code efficiënter en overzichtelijker wordt. [OPGAVE 1] OK. Merk op dat jullie met length telkens over de resterende lijst aan het lopen zijn wat niet efficiënt is. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt: fromDec :: [Int] -> Int fromDec = foldl ((+) . (*10)) 0[OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] De functie numbers staat ook toe dat de letter die net er buiten valt ook gebruikt mag worden. Bijvoorbeeld: de geldige tekens in grondtal 11 zijn 0 t/m 9 en 'a' maar de 'b' wordt ook toegestaan. Dit komt omdat jullie <= 65 + grond - 10 doen i.p.v. < 65 + grond - 10 wanneer jullie het character vergelijken. Overigens had deze functie ook als een combinatie van

filter (om de geldige woorden op te sporen) en map (om

het tupel te construeren) geschreven kunnen worden. [OPGAVE 8] Niet gemaakt. [OPGAVE 9] OK. [OPGAVE 10] OK. mfrancke-jfidijks Nakijker: svisser Cijfer: 7.7 Omschrijving: [ALGEMEEN] Er wordt goed gebruik gemaakt van de standaardfuncties waardoor de functies kort en leesbaar zijn. Ook de layout van de functies ziet er verzorgd uit. Commentaar bij de functies ontbreekt maar in veel gevallen 'beschrijft de code zichzelf' (hoewel een korte beschrijving van een functie geen kwaad kan, zoals bij de functie tel). [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. Er leiden meerdere wegen naar Rome natuurlijk :-) [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] De functie numbers controleert niet of een woord geldig is in het gegeven grondtal. Om het voorbeeld uit de opgave te gebruiken: het woord "faces" is niet geldig in grondtal 16 aangezien de 's' geen geldig teken is (bij grondtal 16 zijn alleen de getallen en 'a' t/m 'f' toegestaan). [OPGAVE 8] OK. [OPGAVE 9] De functie lookAndSay moet al vrij gauw het werk staken aangezien jullie Ints gebruiken om de deelresultaten in op te slaan. De getallen die opgeslagen moeten worden, passen al gauw niet meer in een Int. Hierdoor wordt er geen oneindige lijst met getallen geproduceerd. Het was beter om met lijsten te werken aangezien die wel kunnen doorgroeien. Ook werkt de functie lookAndSay niet voor 0 als begingetal. De lijst zou dan als volgt moeten beginnen: ["0","10","1110","3110","132110","1113122110",... De gekozen aanpak werkt verder wel. Overigens, foldr (:) "" toegepast op een String geeft weer dezelfde String dus dit beïnvloedt de uitkomst van de functie intToString niet. [OPGAVE 10] OK. Nakijker: svisser Cijfer: 7.7 raspauwe Omschrijving:

[ALGEMEEN] De code ziet er net uit en de functies zijn ook kort wat een indicatie kan zijn dat er een goede aanpak is gevolgd. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. Wat ook kan is eerst de tekens omzetten naar de bijbehorende integerwaarde (dus 15 bij f in 16-tallig stelsel) en vervolgens het antwoord uitrekenen. Op die manier hoef je geen onderscheid te maken tussen de stelsels. [OPGAVE 6] OK. [OPGAVE 7] De functie numbers staat voor hogere grondtallen ook toe dat de letter die net er buiten valt ook gebruikt mag worden. Bijvoorbeeld: de geldige tekens in grondtal 11 zijn 0 t/m 9 en 'a' maar de 'b' wordt ook toegestaan. Dit komt omdat je <= n doet i.p.v. < n wanneer je met het grondtal vergelijkt. [OPGAVE 8] Wel een begin mee gemaakt maar niet voltooid. Het gebruik van een Int om het resultaat in op te slaan, zorgt er snel voor dat de waarden niet meer te bewaren zijn (het past simpelweg niet). Het was daarom beter geweest om lijsten te gebruiken voor het 'beschrijfproces' aangezien die wel kunnen doorgroeien. Deze kunnen vervolgens omgezet worden naar de Strings die in het eindresultaat terechtkomen. Verder werkt de gebruikte aanpak wel. [OPGAVE 10] OK. Cijfer: 7.7 pjwjanse-jjvisser Nakijker: svisser Omschrijving: [ALGEMEEN] De code bestaat uit korte functies en heeft een verzorgde layout. Het werkt ook naar behoren. [OPGAVE 1] OK. Merk wel op dat jullie in de functie fromInt met length telkens over de rest van de lijst aan het lopen zijn. Dit maakt de functie minder efficiënt. Hieronder bijvoorbeeld de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt: fromDec :: [Int] -> Int fromDec = foldl ((+) . (*10)) 0[OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] De functie werkt niet als men "10" meegeeft, dit is namelijk gelijk aan het grondtal (dus fromBase 16 "10" zou

16 moeten opleveren maar geeft -1). De functies enumerate en denumerate zijn in feite gelijk aan respectievelijk map normalise en map denormalise. [OPGAVE 6] OK. [OPGAVE 7] OK. Merk op dat deze functie ook geschreven kan worden als een combinatie van filter (voor het opsporen van de geldige woorden) en map (voor het maken van een tupel). [OPGAVE 8] OK. [OPGAVE 9] Deze opgave is niet voltooid. Uit het commentaar maak ik op dat er wel enkele testversies geschreven zijn.. het kan zinvol zijn om dat wel in te leveren aangezien er nu alleen een klein begin staat met undefined op enkele plekken. [OPGAVE 10] OK. De functie keith had ook geschreven kunnen worden als: keith xs = filter keithFilter xs Maar dan kunnen we keithGetallen ook gelijk schrijven als: keithGetallen :: [Int] keithGetallen = filter keithFilter [10..] Cijfer: 7.8 hmpaasse Nakijker: rkoot Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is onvoldoende becommentariëerd. [OPGAVE 1] Prima. [OPGAVE 2] Prima. [OPGAVE 3] Prima. [OPGAVE 4] Prima. [OPGAVE 5] Prima. [OPGAVE 6] Prima. [OPGAVE 7] Werkt niet. [OPGAVE 8] Prima, efficiente implementatie. [OPGAVE 9] Werkt niet. [OPGAVE 10] Prima. mrvaarti-rwerken Nakijker: svisser Cijfer: 7.8

Omschrijving: [ALGEMEEN] De meeste opgaven werken naar behoren alleen bij opgave 7 en 8 ontbreekt er enige functionaliteit (probeer de opgave goed te bestuderen). Er worden lambda-expressies gebruikt waar dat mogelijk is en ook de standaardfuncties worden benut waardoor veel functies uit slechts 1 regel bestaan. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] De functie numbers controleert niet of een woord wel geldige tekens bevat voor het grondtal. Om het voorbeeld uit de opgave te gebruiken: het woord "faces" is niet geldig in grondtal 16 omdat de 's' geen geldig teken is (bij grondtal 16 zijn alleen de getallen en 'a' t/m 'f' toegestaan). [OPGAVE 8] De functie werkt alleen voor de binaire Gray-codering (zoals in het voorbeeld) maar hij werkt niet voor andere grondtallen. Bij hogere grondtallen geeft de functie toGray namelijk waarden die niet voldoen aan de eigenschap die een Gray-codering zou moeten hebben. Zo levert toGray 10 7 de string "4" terwijl toGray 10 8 de string "12" oplevert waardoor er meer dan 1 teken verandert. [OPGAVE 9] OK. [OPGAVE 10] OK. dgerritz Nakijker: rkoot Cijfer: 7.9 Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentariëerd. [OPGAVE 1] Prima. [OPGAVE 2] Prima. [OPGAVE 3] Dit had efficienter geimplementeerd kunnen worden zonder gebruik te maken van machtsverheffen. [OPGAVE 4] Prima. [OPGAVE 5] Prima. [OPGAVE 6] Prima. [OPGAVE 7]

Prima.

[OPGAVE 8]		
[OPGAVE 9] Prima.		
[OPGAVE 10] Prima.		
ltbinsbe	Nakijker: pdwalt	Cijfer: 8
Omschrijving: [ALGEMEEN] + nette code met veel commentaar. prima layout. + gebruik van standaardfuncties. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt. [OPGAVE 5] + werkt. [OPGAVE 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] + werkt. [OPGAVE 8] - ontbreekt. [OPGAVE 9]	Nakijker: pdwalt	Cijfer: 8
+ werkt. [OPGAVE 10] + werkt. + mooie oneindige lijst.		
sjavisse	Nakijker: bschuur	Cijfer: 8.0
Omschrijving: +De gray-codering is efficiënt opgelost. +De look-and-say reeksen zijn oneindig opgebouwd. +De Keith getallen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p. +Het commentaar is goed en effectief. Je zou meer gebruik kunnen maken van patroonherkenning(bij gebruik van meer hogere orde functies aan te raden, deze beide	voorbeeld op lijsten, dan hoef je niet steeds head en ta	
tmsoetho	Nakijker: bschuur	Cijfer: 8.0
Omschrijving: +De look-and-say reeksen zijn oneindig opgebouwd. +De Keith getallen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie). +De code-stijl is netjes. -Er mocht her en der wat meer commentaar bij. Netjes gewerkt. Het opsplitsen in meerdere files is goed en overzichtelijk. Helaas filtert je numbers functie de foute getallen er niet uit, had je de opdracht verkeerd begrepen?		
mvensela-merboxel	Nakijker: svisser	Cijfer: 8.1
Omschrijving: [ALGEMEEN] De code werkt naar behoren op enkele zaken na. Er wordt goed gebruik gemaakt van hulpfuncties om het probleem op te delen en de functies zijn kort en daardoor overzichtelijk. Er wordt vrij vaak gebruik gemaakt van if then else terwijl dit soms ook functioneler kan, bijvoorbeeld met de standaardfunctie filter of met guards. Dit levert ook mooier Haskell op :-) De functies chr en ord (uit de module Char) kunnen gebruikt worden om een Char om te zetten naar een Int en andersom. De functies fromEnum en toEnum werken ook maar geven mit goed weer wat er bedoeld wordt.	nder	

[OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] Als het resultaat van de functie toBase de letter 'a' zou moeten bevatten dan gaat het mis. De bijbehorende Int-waarde van 'a' is 10 en jullie doen (y > 10) in maakLijst' waardoor voor dit geval v + 48 wordt uitgerekend. Maar dit levert een dubbele punt als teken op terwijl het de letter 'a' zou moeten zijn. [OPGAVE 7] OK. Merk op dat dit ook geschreven kan worden als een combinatie van filter (om de geldige woorden op te sporen) en map (om de het tupel te produceren). [OPGAVE 8] De functie grayToInt werkt niet voor "0" maar verder werkt het wel. De waarde van y is dan gelijk aan a $^{\circ}0 + 0 = 1$ waardoor er wordt gekeken of "1" == "0" en daarna blijft de functie verder zoeken zonder iets te vinden. [OPGAVE 9] Jullie gebruiken wel lijsten om de deelresultaten in op te slaan maar door het telkens om te zetten naar een Int gaat het snel mis want de getallen passen dan niet meer in een Int. lookAndSay 1 produceert bijvoorbeeld: ["1","11","21","1211","111221","312211" ,"13112221","1113213211","1288169813"," terwijl de functie lookAndSay' het wel aankan, zie: lookAndSay' (toDec 1113213211) [3,1,1,3,1,2,1,1,1,3,1,2,2,1]lookAndSay produceert dus ook een eindige lijst i.p.v. een oneindige lijst. [OPGAVE 10] OK. jduijn-iduijn Cijfer: 8.2 Nakijker: pdwalt Omschrijving: [ALGEMEEN] + veel commentaar. + redelijk nette code. [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4] + werkt [OPGAVE 5] + werkt [OPGAVE 6] + werkt [OPGAVE 7] - erg onhandige implementatie van numbers; bij grote waarden gaat het programma over z'n nek. beter is om hier te filteren op "woorden" die in je "alfabet"

(afhankelijk van base dus) voorkomen. *Getallen> numbers 22 ["zzzzzzzzzzz"] *** Exception: stack overflow + voor kleine waarden gaat het goed. [OPGAVE 8] + werkt + redelijk efficient [OPGAVE 9] + werkt + oneindige lijst [OPGAVE 10] + werkt + oneindige lijst hlversto Nakijker: rkoot Cijfer: 8.2 Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Niet veel gebruik gemaak van standaard- en hulpfuncties. Code is onvoldoende becommentariëerd. [OPGAVE 1] Prima. [OPGAVE 2] Prima. [OPGAVE 3] Prima. [OPGAVE 4] Prima. [OPGAVE 5] Prima. [OPGAVE 6] Prima. [OPGAVE 7] Prima. [OPGAVE 8] Werkt niet. [OPGAVE 9] Prima. [OPGAVE 10] Prima. mahashi-mjhobbel Nakijker: svisser Cijfer: 8.2 Omschrijving: [ALGEMEEN] Over het algemeen een goed practicum: de functies werken naar behoren en er is aandacht besteed aan de layout van de functies. Op bepaalde plekken kan er handig gebruik gemaakt worden van standaardfuncties. Het is niet fout om het zelf te schrijven maar het gebruik van standaardfuncties maakt de structuur wel duidelijker. Het is ook gebruikelijk om de types bij de functie zelf te zetten en niet bij elkaar in een lijst. [OPGAVE 1] OK. Merk op dat jullie met init en last telkens over de resterende lijst aan het lopen zijn wat niet efficiënt is. Hieronder de functie fromDec

met een foldl waarmee men maar 1 keer over de lijst loopt:

fromDec :: [Int] -> Int fromDec = foldl ((+) . (* 10)) 0

[OPGAVE 2]

OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] OK. Het werk wat in deze functie gedaan wordt, bestaat eigenlijk uit het filteren van elementen uit een lijst en vervolgens het toepassen van een functie. Je had deze functie dus ook kunnen schrijven met filter en map: numbers2 a xs = map tupel (filter ok xs)where ok x =and (map (|tupel y = (y, fromBase a y)|Of met lijstcomprehensie: numbers3 a xs = [(x, fromBase a x) | x <- xs, ok x]where ok x =and (map ([OPGAVE 8] OK. Bij deze opgave (en bij de Keith-opgave) wordt er vaak (x:xs) gebruikt terwijl de functies zelf x en xs niet los gebruiken. Je kunt dus ook de hele lijst een naam geven, zoals jullie wel doen bij andere opgaven. [OPGAVE 9] De functie gaat na de eerste paar elementen de fout in omdat de functie look een verkeerd resultaat geeft. Zo geeft look [1,2,1,1] de waarde "3112" terwijl dit "111221" zou moeten zijn, dat is namelijk een beschrijving van de lijst: 1x een 1, 1x een 2 en 2x een 1. Wat jullie doen is het tellen van het aantal keer dat de 1 voorkomt in de hele lijst en jullie zetten dat vervolgens vooraan (wat inderdaad 3x is). Wat je eerst zou moeten doen is de lijst opdelen in opeenvolgende stukjes getallen om vervolgens de lengte van elk stukje te bepalen. Je krijgt dan de gevraagde beschrijving van een lijst. [OPGAVE 10] OK. Ook hier is de structuur van een filter zichtbaar (in keithGetallen en keithGetallen"). Er bestaat ook een standaardfunctie elem die vertelt of een element voorkomt in een lijst (bijv. elem 3 [1,2,3] geeft True). Men zou dus kunnen schrijven: keithGetallen = filter isKeith [10..] where is Keith x = elem x (keith Reeks x) Cijfer: 8.5 rtharder Nakijker: bschuur Omschrijving: +De look-and-say reeksen zijn oneindig opgebouwd. +De Keith getallen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie). +Het commentaar is goed en effectief. +De code-stijl is netjes. tsteemer Nakijker: bschuur Cijfer: 8.5

Omschrijving:

+De gray-codering is efficiënt opgelost maar loopt vast bij bepaalde randgevallen, ik kon niet achterhalen waar de fout zit.

+De look-and-say reeksen zijn oneindig opgebouwd maar lopen gauw tegen de limiet van integer aan. +De Keith getallen zijn oneindig opgebouwd.

+De code-stijl is netjes.

+Het commentaar is goed en effectief.

Je hebt al een aantal higher-order functies gebruikt maar dit kan veel meer. Vooral bij de eerste opgaven. Je hebt netjes correct gewerkt, je had her

tromberg-rjanssen		Nakijker: bschuur	Cijfer: 8.5
Omschrijving: -De gray-codering is door de recursieve aanpak +De look-and-say reeksen zijn oneindig opgebo +De Keith getallen zijn oneindig opgebouwd. +Je hebt goed gebruik gemaakt van high order f +De code-stijl is netjes.	uwd.		
Doordat de graycode functie een oneindige lijst numbers functie filtert foute woorden er niet uit.		nag, je had hier beter een directe formula kunnen prol pegrepen?	peren te vinden. Je
whustinx		Nakijker: bschuur	Cijfer: 8.5
+De Keith getallen zijn oneindig opgebouwd. +De look-and-say reeksen zijn oneindig opgebo +Je hebt goed gebruik gemaakt van high order f +De code-stijl is netjes, maar vooral de naamge -Commentaar is volledig afwezig.	nuwd. Tunctions. ving is naar het einde toe o	o' fouten. Hier moet misschien ergens een randgeval eryptisch.	afgevangen worden.
In je numbers functie zit een subtiele fout: een g naamgeving voor functies gebruikt is commenta		als het grondtal wordt geaccepteerd, dit hoort niet. A	ls je cryptische
rhaan		Nakijker: bschuur	Cijfer: 8.5
voor de numbers functie. De gCfrom functie in	wwd. wat traag en zorgt voor ov =35 voor het grondtal maa grayCode heeft geen basis	verflows. r dit moet n<=36 zijn, dit zorgt voor fouten bij grond geval en geeft fouten, doordat deze functie recursief	
ontstaan stack overflows bij grote getallen en is btdijk	de functie traag.	Nakijker: rkoot	Cijfer: 8.5
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpf Voldoende commentaar. [OPGAVE 1] Prima. [OPGAVE 2] Prima.	functies.		
[OPGAVE 3] Prima.			
OPGAVE 4] Prima.			
[OPGAVE 5] Prima.			
[OPGAVE 6] Prima.			
[OPGAVE 7] Prima.			
OPGAVE 8] Werkt, maar inefficient geimplementeerd.			
[OPGAVE 9] Gaat mis voor het randgeval 0. Te inefficient om de eerste 15 te genereren.			
[OPGAVE 10]			

cjlblom		Nakijker: rkoot	Cijfer: 8.5
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncti Code is goed becommentariëerd.	es.		
[OPGAVE 1] Prima.			
[OPGAVE 2] Prima.			
[OPGAVE 3] Prima.			
[OPGAVE 4] Prima.			
[OPGAVE 5] Prima.			
[OPGAVE 6] Prima.			
[OPGAVE 7] Controleert niet of het woord wel een geldig getal is [("z",35)]).	in de opgegven basis (bijvoorbeeld: numbe	ers 17 ["z"] had de leg	e lijst op moeten leveren, niet
[OPGAVE 8] Niet inverteeerbaar voor bijvoorbeeld: *Getallen> snd (grayCode 11) 121 "1a0" *Getallen> fst (grayCode 11) "1a0" 122			
[OPGAVE 9] Werkt niet voor startwaardes groter dan 9.			
[OPGAVE 10] Prima			
dekuppev		Nakijker: rkoot	Cijfer: 8.5
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncti Code is goed becommentariëerd. [OPGAVE 1]	es.		
Prima. [OPGAVE 2]			
Prima. [OPGAVE 3]			
Prima. [OPGAVE 4]			
Prima.			
[OPGAVE 5] Prima.			
[OPGAVE 6] Prima.			
[OPGAVE 7] Werkt niet altijd correct, bijvoorbeeld op het geval nu	umbers 27 ["ed","ylfcy","iroojpi","otqswf"].		
[OPGAVE 8] Werkt, maar is erg inefficient.			
[OPGAVE 9] Werkt niet voor startwaardes groter dan 9.			

[OPGAVE 10] Prima.		
bspaans	Nakijker: rkoot	Cijfer: 8.5
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentariëerd.	NAKIJKEI. IKOOU	Cijici. 6.3
[OPGAVE 1] Prima.		
[OPGAVE 2] Prima.		
[OPGAVE 3] Prima.		
[OPGAVE 4] Prima.		
[OPGAVE 5] Prima.		
[OPGAVE 6] Prima.		
[OPGAVE 7] Prima.		
[OPGAVE 8] Het randgeval 0 gaat fout.		
[OPGAVE 9] Te inefficient om de eerste 15 te genereren.		
[OPGAVE 10] Te inefficient om de eerste 20 te genereren.		
nroumimp	Nakijker: svisser	Cijfer: 8.7
Omschrijving: [ALGEMEEN] De code is bij de meeste opgaven kort wat een indicatie kan z mogelijk is). Het werkt ook naar behoren.	ijn dat er een goede aanpak is gekozen (zoals het gebr	uik van een foldfunctie waar dat
[OPGAVE 1] OK.		
[OPGAVE 2] OK.		
[OPGAVE 3] OK.		
[OPGAVE 4] OK.		
[OPGAVE 5] OK.		
[OPGAVE 6] OK.		
[OPGAVE 7] OK.		
Hier had je enkele standaardfuncties handig kunnen gebruiken om de goede woorden uit de lijst te halen en vervolgens map om te zetten naar een tupel.		

Voor foldr (&&) True bestaat overigens ook een standaardfunctie, deze heet 'and' en is precies zo gedefinieerd: and = foldr (&&) True. Evenzo is er de functie 'or': or = foldr (||) False.

[OPGAVE 8]

Deze opgave is niet volledig gemaakt; ook is de code vrij omslachtig waarmee niet voor de meest handige Graycodering is gekozen. Misschien dat je in commentaar je aanpak had kunnen toelichten? Sommige Graycoderingen zijn namelijk wel goed te vangen met een terugkerende structuur. [OPGAVE 9] OK. [OPGAVE 10] OK. ompennin Nakijker: svisser Cijfer: 8.7 Omschrijving: [ALGEMEEN] De code werkt over het algemeen naar behoren. De code is wel erg breed met lange commentaarregels en een brede layout bij functies. Een where hoeft niet onder de = van de vorige regel te staan maar kan ook eerder beginnen. Guards kunnen ook al eerder geplaatst worden, dus bijvoorbeeld: naamVanFunctie langeParameter nogEenParamater || langeParameter > 10 = ...|| otherwise = ... Probeer de functies beschrijvende namen te geven want 'aap' bij verscheidene opgaven komt niet echt serieus over. Ook de leesbaarheid van de code komt het niet ten goede. [OPGAVE 1] OK. De overeenkomsten tussen de eerste aantal opgaven worden mooi verzorgd door algemene functies die het werk verrichten voor een gegeven grondtal. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] Leuk beargumenteerd maar een functie laten crashen omdat de invoer onjuist is, druist wel tegen de principes van software-ontwikkeling in. Het verhelpen is niet zo omslachtig als beweerd wordt, zo kun je met filter eerst de geldige woorden voor een gegeven base eruit halen en vervolgens deze omzetten: map tupel (filter ok words) where ok :: String -> Bool tupel :: String -> (String, Int) tupel x = ...[OPGAVE 8] OK, en nog met een efficiënte aanpak ook. [OPGAVE 9] OK. [OPGAVE 10] OK. reahoef-irrencke Nakijker: svisser Cijfer: 8.7 Omschrijving: [ALGEMEEN]

De opgaven worden netjes opgedeeld in hulpfuncties waarmee de oorspronkelijke functies handig geschreven kunnen worden. Hierdoor blijven de functies ook kort.

De code komt overzichtelijk over omdat jullie, als een van de weinige, geheel binnen de 80-tekens limiet blijven (voor maximale breedte) die men wel hanteert om broncode leesbaar te houden.

[OPGAVE 1]

OK.

Merk op dat jullie met length telkens over de resterende lijst aan het lopen zijn wat niet efficiënt is. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt:

fromDec :: [Int] -> Int

fromDec = foldl ((+) . (*10)) 0

[OPGAVE 2]

OK.

[OPGAVE 3]

OK.

[OPGAVE 4]

OK.

[OPGAVE 5]

OK.

[OPGAVE 6]

OK.

Voor het omzetten van een Char naar een Int en andersom bestaan de functies ord en chr in de module Char. De functies toEnum en fromEnum werken ook maar geven minder goed weer wat er bedoeld wordt.

[OPGAVE 7]

OK.

Bij de functie stringToInt hadden jullie de functie charToInt kunnen gebruiken want het gedeelte met guards lijkt bijzonder veel op elkaar. Ook had hier handig gebruik gemaakt kunnen worden van enkele standaardfuncties, zoals filter (om de geldige woorden op te sporen) en map (om het tupel te produceren).

[OPGAVE 8]

De functie fromGray is niet gedefinieerd voor "0". Het volgende geeft bijvoorbeeld een foutmelding (gebeurt ook bij andere grondtallen):

let g = grayCode 10 in map ((fst g).(snd g)) [0..10]

[*** Exception: Bestaat de code bij het gekozen grondgetal?

[OPGAVE 9]

De functie lookAndSay is niet gedefinieerd voor 0. Het levert dan een lijst met lege Strings op terwijl het zou moeten beginnen met

["0","10","1110","3110","132110","1113122110"

[OPGAVE 10]

OK.

In plaats van een lijst terug te geven (en later concat om de lege lijsten weg te werken), hadden jullie ook filter kunnen gebruiken met een functie isKeithGetal als predicaat om te testen. Dit scheelt weer in het werk dat verricht moet worden.

jkoperdr Nakijker: pdwalt Cijfer: 8.8

Omschrijving:

[ALGEMEEN]

+ mooi codestijl; beetje commentaar maar vooral mooie layout en veelvuldig gebruik van standaardfuncties.

[OPGAVE 1]

+ werkt

[OPGAVE 2]

+ werkt

[OPGAVE 3]

+ werkt

[OPGAVE 4]

+ werkt

[OPGAVE 5] + werkt [OPGAVE 6] + werkt [OPGAVE 7] + werkt wel, maar geeft helaas geen resultaat als het "getal" gro [OPGAVE 8] - helaas, foutje, successors zijn niet op 1 plek verschillend: *Getallen> (snd\$grayCode 5) 615 "4430" *Getallen> (snd\$grayCode 5) 614 "4424" omdat grayCode stiekem to/fromBase is. [OPGAVE 9] + werkt [OPGAVE 10] + werkt gloupias	oter is dan MAX_INT Nakijker: rkoot	Cijfer: 8.8
Omschrijving:		
[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is voldoende becommentariëerd.		
[OPGAVE 1] Dit had efficienter geimplementeerd kunnen worden zonder ge	ebruik te maken van machtsverheffen.	
[OPGAVE 2] Prima.		
[OPGAVE 3] Prima.		
[OPGAVE 4] Prima.		
[OPGAVE 5] Prima.		
[OPGAVE 6] Prima.		
[OPGAVE 7] Prima.		
[OPGAVE 8] Werkt, maar niet erg efficient.		
[OPGAVE 9] Werkt niet indien de startwaarde al groter is dan 9.		
[OPGAVE 10] Prima.		
gdijkstr-rjhensin	Nakijker: rkoot	Cijfer: 8.8
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is matig becommentariëerd.		
[OPGAVE 1] Prima.		
[OPGAVE 2] Prima.		
[OPGAVE 3] Prima.		
[OPGAVE 4] Prima.		
[OPGAVE 5] Prima.		

[OPGAVE 6] Prima. [OPGAVE 7] Prima. [OPGAVE 8] Jullie graycode is niet altijd inverteerbaar, bijvoorbeeld voor het getal 100 met radix 10. [OPGAVE 9] Prima. [OPGAVE 10] Prima. rawagenm Nakijker: svisser Cijfer: 8.8 Omschrijving: [ALGEMEEN] Er wordt gebruik gemaakt van standaardfuncties waar dat mogelijk is en op enkele details na werken de functies naar behoren. [OPGAVE 1] OK. Merk op dat je met length telkens over de resterende lijst aan het lopen bent wat niet efficiënt is. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt: fromDec :: [Int] -> Int fromDec = foldl ((+) . (*10)) 0[OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. Voor het omzetten van Char naar Int en andersom bestaan de functies ord en chr in de module Char. De functies fromEnum en toEnum werken ook maar geven minder weer wat er bedoeld wordt. [OPGAVE 6] OK. [OPGAVE 7] De functie numbers staat ook toe dat de letter die net er buiten valt ook gebruikt mag worden. Bijvoorbeeld: de geldige tekens in grondtal 11 zijn 0 t/m 9 en 'a' maar de 'b' wordt ook toegestaan. Dit komt omdat je map ((<=g).charToInt) i.p.v. map ((woorden. [OPGAVE 8] OK. [OPGAVE 9] De functie werkt niet voor een startgetal van 36 en hoger. De functie intToChar is voor die waarden niet gedefinieerd. Ook is het omzetten naar een letter niet de bedoeling: als men nu bijvoorbeeld lookAndSay 15 bekijkt dan begint die met: ["f","1f","111f","311f",...] Maar "15" zou het eerste element moeten zijn: ["15","1115","3115","132115","1113122115",...] [OPGAVE 10] OK. sdriel Nakijker: bschuur Cijfer: 9.0

Omschrijving:

- -De gray-codering is door de oneindige aanpak wat traag, een directe formule was beter geweest.
- +Je hebt goed gebruik gemaakt van high order functions. Je had deze ook kunnen toepassen bij de eerste opgaven!
- +Het commentaar is goed en effectief.
- +De code-stijl is netjes.
- +De look-and-say reeksen zijn oneindig opgebouwd.
- +De Keith getallen zijn oneindig opgebouwd.

Je hebt de problemen over het algemeen wat omslachtig aangepakt (bijvoorbeeld met de log functie in de eerste opgaven, dit kan simpeler en netter met rem en div). Je number functie bevat een kleine fout waarbij een letter met waarde gelijk aan het grondtal alsnog als correct wordt gezien(dit moet altijd eentje minder zijn). Je stopt veel functionaliteit 'in-line' terwijl je bijvoorbeeld voor het converteren van een letter naar een getal en viseversa een goed afgesplitste functie kunt schrijven. Naar het einde toe maak je steeds meer gebruik van hogere orde functie, goed! Je hebt verder goed correct gewerkt, super!

rgroot Nakijker: bschuur Cijfer: 9.0

Omschrijving:

- +De gray-codering is zonder oneindige lijsten opgelost.
- +De look-and-say reeksen zijn oneindig opgebouwd.
- +De Keith getallen zijn oneindig opgebouwd.
- +De code-stijl is netjes.
- +Het commentaar is goed en effectief.
- -Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie).

Opgave 6:

Je validBase functie is niet helemaal correct, deze neemt het g-de getal ook als valide, onterecht(je zit er dus aan de bovengrens steed 1 naast). validBase 16 "fg1" zal true opleveren, ondanks dat de g er buiten zou moeten vallen.

fsteeg-hkbarnev Nakijker: rkoot Cijfer: 9.0

Omschrijving:

[ALGEMEEN]

Code is duidelijk leesbaar.

Handig gebruik gemaak van standaard- en hulpfuncties.

Code is goed becommentariëerd.

[OPGAVE 1]

Prima

[OPGAVE 2]

Prima.

[OPGAVE 3]

Prima.

[OPGAVE 4]

Prima.

[OPGAVE 5]

Prima.

[OPGAVE 6]

Prima.

OPGAVE 7

Werkt niet altijd correct, bijvoorbeeld voor numbers 18 ["foqnae","i","hts","xgvb","tkcmd","erhmty","I","qdoooq"]

[OPGAVE 8]

Werkt niet altijd correct, bijvoorbeeld voor het getal 216 in radix 8.

[OPGAVE 9]

Prima.

[OPGAVE 10]

Prima.

mlmbroer-jmwbrete Nakijker: svisser Cijfer: 9.1

Omschrijving:

[ALGEMEEN]

Prima practicum, alleen de aanpak bij vraag 9 leidt er toe dat er verkeerde resultaten worden gegeven. De code ziet er netjes uit,

er is aandacht besteed aan de layout van de functies en er worden

standaardfuncties gebruikt waar dat mogelijk is.

[OPGAVE 1]

OK.

Merk op dat je met init en last telkens over de resterende lijst aan het lopen bent wat niet efficiënt is. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt: fromDec :: [Int] -> Int from Dec = foldl ((+) . (* 10)) 0[OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] OK. [OPGAVE 8] OK, prima aanpak. [OPGAVE 9] De functie werkt wel maar moet al vrij gauw het werk staken aangezien de functie read het niet meer aankan. Als gevolg hiervan wordt er geen oneindige lijst geconstrueerd. Het was beter geweest om met lijsten te werken want een Int is beperkt qua omvang terwijl lijsten wel kunnen doorgroeien. [OPGAVE 10] OK. Cijfer: 9.2 mkroese Nakijker: svisser Omschrijving: [ALGEMEEN] Prima practicum, er wordt gebruik gemaakt van de mogelijkheden die Haskell biedt, zoals standaardfuncties, lijstcomprehensies e.d. De code werkt ook zoals het moet, alleen de helft van de Gray-opgave ontbreekt. [OPGAVE 1] OK. [OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] OK. [OPGAVE 8] De functie fromGray is niet geschreven. [OPGAVE 9] OK. [OPGAVE 10] OK.

Nakijker: svisser Cijfer: 9.2 maooster Omschrijving: [ALGEMEEN] De code ziet er overzichtelijk uit en er wordt gebruik gemaakt van de mogelijkheden die Haskell biedt, zoals lijstcomprehensies, standaardfuncties e.d. Er worden ook hulpfuncties gedefinieerd waar dat enigzins kan. [OPGAVE 1] OK. Met length loop je telkens over de lijst heen wat deze implementatie minder efficiënt maakt. Hieronder de functie fromDec met een foldl waarmee men maar 1 keer over de lijst loopt: fromDec :: [Int] -> Int fromDec = foldl ((+) . (*10)) 0[OPGAVE 2] OK. [OPGAVE 3] OK. [OPGAVE 4] OK. De functie eerste bestaat al als standaardfunctie, deze heet fst. [OPGAVE 5] OK. [OPGAVE 6] OK. [OPGAVE 7] OK. [OPGAVE 8] In de code geef je aan een probleem te hebben opgemerkt. De functie voor oneven grondtallen werkt niet voor het "0"-geval maar het werkt wel voor de overige waarden. Bijvoorbeeld: let t = grayCode 33 in map (fst t) (map (snd t) [1..10])(geeft [1..10]) let t = grayCode 33 in map (fst t) (map (snd t) [0..10])(geeft geen resultaat) De functie voor even getallen werkt wel: let s = grayCode 14 in map (fst s) (map (snd s) [1..10])(geeft [1..10]) let s = grayCode 14 in map (fst s) (map (snd s) [0..10])(geeft [0..10]) De code werkt verder wel. [OPGAVE 9] OK. [OPGAVE 10] OK. Je had ook de getallen onder de 10 eruit kunnen filteren door de lijst te laten beginnen bij 10 (dus: [10..]), dan is die extra controle niet nodig ;-) lsstoel Cijfer: 9.3 Nakijker: pdwalt Omschrijving: [ALGEMEEN] + mooie code met veel commentaar dat bovendien leesbaar is. [OPGAVE 1]

+ werkt. [OPGAVE 2]

```
+ werkt.
[OPGAVE 3]
+ werkt.
[OPGAVE 4]
+ werkt.
[OPGAVE 5]
+ werkt.
[OPGAVE 6]
+ werkt.
[OPGAVE 7]
+ werkt.
[OPGAVE 8]
 - decoderen werkt niet, maar dat is al aangegeven.
+ encoderen werkt wel.
+ efficient.
[OPGAVE 9]
+ werkt.
+ mooie oneindige lijst
[OPGAVE 10]
+ werkt.
+ mooie lijstconstructie.
                                                                                                                                   Cijfer: 9.3
pqgroot
                                                                                            Nakijker: svisser
Omschrijving:
[ALGEMEEN]
De code bestaat uit korte functies en er wordt goed gebruik gemaakt
van de standaardfuncties die in verscheidene modules beschikbaar zijn.
Ook werkt het naar behoren.
[OPGAVE 1]
OK.
Merk op dat je met length telkens over de resterende lijst
aan het lopen bent wat niet efficiënt is. Hieronder de functie fromDec
met een foldl waarmee men maar 1 keer over de lijst loopt:
fromDec :: [Int] -> Int
fromDec = foldl ((+) . (*10)) 0
[OPGAVE 2]
OK.
[OPGAVE 3]
OK.
[OPGAVE 4]
OK.
[OPGAVE 5]
OK.
[OPGAVE 6]
OK.
[OPGAVE 7]
OK.
De functies toGrayCode en fromGrayCode sluiten niet helemaal
op elkaar aan. Voor elk getal n zou de waarde die toGrayCode oplevert
weer moeten kunnen worden ingelezen door fromGrayCode om n op te
leveren maar dit gebeurt niet voor n <= base.
Vergelijk bijvoorbeeld:
togray :: [String]
togray = map (toGrayCode 10) [0..20]
met:
fromgray :: [Int]
fromgray = map (fromGrayCode 10) togray
fromgray zou [0..20] moeten opleveren maar begint eerst met aftellen
vanaf 9 (in het algemeen: base - 1).
Ook ontbrak de hoofdfunctie grayCode hoewel deze eenvoudig geschreven
```

kan worden als combinatie van toGrayCode en fromGrayCode	le.	
[OPGAVE 9] OK.		
[OPGAVE 10] OK.		
Probeer gemeenschappelijke expressies in een where te zetten, maakt de functies efficiënter en ook korter. De expressie sum(snd(splitAt ((length i)-j) i)) wordt bijvoorbeeld vier keer gebruikt in isKeith.		
Je had hier ook een filter kunnen gebruiken in plaats van de functie kg.		
vrbons-hckampma	Nakijker: bschuur	Cijfer: 9.5
Omschrijving: +De gray-codering is efficiënt opgelost. +De look-and-say reeksen zijn oneindig opgebouwd, maar is become te to the settle getallen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p.) +De code-stijl is netjes. +Het commentaar is goed en effectief.	p.v. recursie, lijstcomprehensie).	
	code is al netjes compact, met hogere orde functies kan je dit nog verder	
wlelsing	Nakijker: bschuur	Cijfer: 9.5
Omschrijving: -De gray-codering is door de recursieve aanpak wat traag en z +De look-and-say reeksen zijn oneindig opgebouwd. +De Keith getallen zijn oneindig opgebouwd. +Je hebt goed gebruik gemaakt van high order functions. +De code-stijl is netjes.		
	meer kunnen gebruiken voor de eerste opgave om die nog net iets strakke	
ergallo Omschrijving:	Nakijker: rkoot	Cijfer: 9.5
[ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentariëerd.		
[OPGAVE 1] Prima.		
[OPGAVE 2] Prima.		
[OPGAVE 3] Prima.		
[OPGAVE 4] Prima.		
[OPGAVE 5] Prima.		
[OPGAVE 6] Prima.		
[OPGAVE 7] Prima.		
[OPGAVE 8] Je implementatie is correct, maar helaas nogal inefficient.		
[OPGAVE 9] Prima.		
[OPGAVE 10]		
Prima.		
	Nakijker: svisser	Cijfer: 9.5
Prima. mgrimme Omschrijving:	Nakijker: svisser	Cijfer: 9.5

[ALGEMEEN]

De code werkt naar behoren maar er wordt bij tijden wel een imperatieve aanpak gebruikt terwijl er ook een 'functionelere' aanpak bestaat (zie bijvoorbeeld bij opgave 1). De code is wel netjes en bestaat uit korte functies.

Hulpfuncties die voor de rest van de code niet relevant zijn, kun je in een where zetten zodat alleen de functie waar ze bij horen ze kan gebruiken.

[OPGAVE 1]

OK.

De functie fromDec met een foldl:

fromDec :: [Int] -> Int

|fromDec = foldl ((+) . (* 10)) 0

[OPGAVE 2]

OK.

[OPGAVE 3]

OK.

[OPGAVE 4]

OK.

[OPGAVE 5]

OK.

[OPGAVE 6]

OK.

[OPGAVE 7]

De functie numbers controleert niet of de meegegeven woorden wel geldig zijn in het grondtal. Om het voorbeeld uit de opgave te gebruiken: het woord "faces" is niet geldig in grondtal 16 aangezien de 's' geen geldig teken is (bij grondtal 16 zijn alleen de getallen en 'a' t/m 'f' toegestaan).

Deze functie kan ook worden geschreven als een combinatie van filter (om de geldige woorden uit de lijst de halen) en map (om elk woord om te zetten naar een tupel).

[OPGAVE 8]

OK.

De functies werken voor de meeste invoerwaarden wel maar voor hele grote getallen zijn er enkele gevallen waar de Gray-eigenschap niet meer geldt. Vergelijk bijvoorbeeld:

let s = grayCode 24 in (snd s) 291504127

(geeft: "1beejag")

let s = grayCode 24 in (snd s) 291504128 (geeft: "011beejaf")

[OPGAVE 9]

OK.

Ter observatie: de functie sayStrings kan ook worden geschreven als een combinatie van map en concat:

sayStrings ss = concat (map sayString) ss

Er bestaat ook een standaardfunctie concatMap die dit combineert:

sayStrings ss = concatMap sayString ss

Aangezien ss nu bij beide kanten aan de rechterkant staat mag het ook worden weggelaten:

sayStrings = concatMap sayString

[OPGAVE 10]

OK. affboth-lveerman Nakijker: rkoot Cijfer: 9.6 Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Commentaar is spaarzaam gebruikt. fromBase: Een oplossing zonder gebruik te maken van machtsverheffen was efficiënter geweest. [OPGAVE 2] Prima. [OPGAVE 3] Prima. [OPGAVE 4] Prima. [OPGAVE 5] Prima. [OPGAVE 6] Prima. [OPGAVE 7] Prima. [OPGAVE 8] Jullie gray-codering is niet inverteerbaar indien het grondgetal gelijk is aan het getal zelf. [OPGAVE 9] Prima. [OPGAVE 10] Prima. imberg-cwbbonen Nakijker: pdwalt Cijfer: 9.7 Omschrijving: [ALGEMEEN] + veel commentaar, nette layout; fijn om naar te kijken. + standaardfuncties gebruikt. [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4] + werkt [OPGAVE 5] + werkt [OPGAVE 6] + werkt [OPGAVE 7] + werkt [OPGAVE 8] + werkt + efficient [OPGAVE 9] + werkt + oneindige lijst [OPGAVE 10] + werkt + oneindige lijst elrenkem-ogrottie Nakijker: rkoot Cijfer: 9.8 Omschrijving: [ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentariëerd. [OPGAVE 1]

Prima.			
[OPGAVE 2] Prima.			
[OPGAVE 3] Prima.			
[OPGAVE 4] Prima.			
[OPGAVE 5] Prima.			
[OPGAVE 6] Prima.			
[OPGAVE 7] Prima.			
[OPGAVE 8] Prima, efficient.			
[OPGAVE 9] Werkt niet voor het randgeval 0.			
[OPGAVE 10] Prima.			
bjliefer-ptpkokke		Nakijker: rkoot	Cijfer: 9.8
Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfunct Code is goed becommentariëerd. [OPGAVE 1] Prima.	ies.		
[OPGAVE 2] Prima.			
[OPGAVE 3] Prima.			
[OPGAVE 4] Prima.			
[OPGAVE 5] Prima.			
[OPGAVE 6] Prima.			
[OPGAVE 7] Prima.			
[OPGAVE 8] Prima, efficiënte implementatie.			
[OPGAVE 9] Werkt niet voor het randgeval 0.			
[OPGAVE 10] Prima.			