

Nakijker View

How to use this page: [reviewer \(studentassistent\)](#), [lecturer](#)

For lecturers: return to the [assignments page](#)

Opgave 1

ybouma	Cijfer: 5.0	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: Hetgeen je gedaan hebt is kloppend en goed geschreven. Je had meer hogere-orde functies kunnen gebruiken om de code netter te krijgen.		
<input type="button" value="Wijzig"/>		

wokat	Cijfer: 7.5	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: +De look-and-say reeksen zijn oneindig opgebouwd maar lopen tegen de beperkingen van een integer aan. +Je hebt goed gebruik gemaakt van high order functions. +De code-stijl is netjes. +Het commentaar is goed en effectief. Er zitten wat subtiele foutje in je werk. Zo heb je bij toBin het basisgeval 0 niet afgevangen wat het programma in een oneindige loop gooit. Verder klopt de numbers functie niet helemaal, de getallen met dezelfde waarde als het grondtal worden nog steeds als kloppend gezien(dit moet 1 minder zijn).		
<input type="button" value="Wijzig"/>		

wlelsing	Cijfer: 9.5	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: -De gray-codering is door de recursieve aanpak wat traag en zorgt voor overflows. +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. Goed werk! Veel higher orde runcties gebruikt, je zou ze nog meer kunnen gebruiken voor de eerste opgave om die nog net iets strakker te maken.		
<input type="button" value="Wijzig"/>		

whustinx	Cijfer: 8.5	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: +De gray-codering is efficiënt opgelost maar geeft helaas 'division by zero' fouten. Hier moet misschien ergens een randgeval afgevangen worden. +De Keith getallen zijn oneindig opgebouwd. +De look-and-say reeksen zijn oneindig opgebouwd. +Je hebt goed gebruik gemaakt van high order functions. +De code-stijl is netjes, maar vooral de naamgeving is naar het einde toe cryptisch. -Commentaar is volledig afwezig. In je numbers functie zit een subtiele fout: een getal met dezelfde waarde als het grondtal wordt geaccepteerd, dit hoort niet. Als je cryptische naamgeving voor functies gebruikt is commentaar		
<input type="button" value="Wijzig"/>		

vrbons-hckampma	Cijfer: <input type="text" value="9.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: <div><div>+De gray-codering is efficiënt opgelost. +De look-and-say reeksen zijn oneindig opgebouwd, maar is beperkt door Int te gebruiken. +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. +Het commentaar is goed en effectief.</div><div>Erg goed gewerkt je hebt een goede en correcte code-stijl. Je code is al netjes compact, met hogere orde functies kan je dit nog verder verbeteren.</div></div> <div>Wijzig</div>		

vcuijre	Cijfer: <input type="text" value="1.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: <div><div>Onvolledig? Look and say was niet helemaal volgens de specificatie maar wel goed en oneindig.</div></div> <div>Wijzig</div>		

tsteemer	Cijfer: <input type="text" value="8.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: <div><div>+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.</div><div>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 en der kunnen kijken of er een kortere oplossing was voor de problemen.</div></div> <div>Wijzig</div>		

tromberg-rjanssen	Cijfer: <input type="text" value="8.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving: <div><div>-De gray-codering is door de recursieve aanpak wat traag en zorgt voor overflows. +De look-and-say reeksen zijn oneindig opgebouwd. +De Keith getallen zijn oneindig opgebouwd. +Je hebt goed gebruik gemaakt van high order functions, vooral in look-and say, je had dit nog meer kunnen toepassen. +De code-stijl is netjes.</div><div>Doordat de graycode functie een oneindige lijst opbouwt wordt deze te traag, je had hier beter een directe formula kunnen proberen te vinden. Je numbers functie filtert foute woorden er niet uit, had je de opdracht goed begrepen?</div></div> <div>Wijzig</div>		

tmsoetho	Cijfer: <input type="text" value="8.0"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
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).
+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?

Wijzig

tleroi

Cijfer: 5.5

Bevestigd: ja Mail

Nakijker: bschuur

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 aangeroepen 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.

Wijzig

taveld

Cijfer: 7.0

Bevestigd: ja Mail

Nakijker: bschuur

Omschrijving:

-Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie).
+De Keith getallen zijn oneindig opgebouwd.
+De look-and-say reeksen zijn oneindig opgebouwd.
-Commentaar is volledig afwezig.

Alhoewel al je to* functies lijken te kloppen kunnen ze niet goed omgaan met grote getallen, ik heb niet kunnen uitvissen waar hem dat in zit(waarschijnlijk beperkingen van int). Je getCharacter functie in toBase bevat een onvolledig patroon. Je numbers functie laat ook alle letters door die gelijk zijn aan het grondtal, dit moeten echter alle letters zijn die eentje minder zijn dan het grondtal.

Wijzig

talles

Cijfer: 4.0

Bevestigd: ja Mail

Nakijker: bschuur

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.

Wijzig

sttimmer

Cijfer: 10

Bevestigd: ja Mail

Nakijker: bschuur

Omschrijving:

+De gray-codering is efficiënt 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.

Wijzig

sjveldhu	Cijfer: <input type="text" value="7.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving:		
<div>+De look-and-say reeksen zijn oneindig opgebouwd, maar gaat op een gegeven moment de fout in (Integer inplaats van Int gebruiken?). +De Keith getallen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie). Zeker bij de eerste opgaven. +De code-stijl is netjes. +Het commentaar is goed en effectief.</div>		
<input type="button" value="Wijzig"/>		

sjavisse	Cijfer: <input type="text" value="8.0"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving:		
<div>+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.v. recursie, lijstcomprehensie). +Het commentaar is goed en effectief.</div> <div>Je zou meer gebruik kunnen maken van patroonherkenning(bijvoorbeeld op lijsten, dan hoeft je niet steeds head en tail aan te roepen). Ook is het gebruik van meer hogere orde functies aan te raden, deze beide kunnen de grootte van je code flink inperken en de leesbaarheid vergroten.</div>		
<input type="button" value="Wijzig"/>		

sdriel	Cijfer: <input type="text" value="9.0"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving:		
<div>-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.</div> <div>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</div>		
<input type="button" value="Wijzig"/>		

sagieske	Cijfer: <input type="text" value="5.0"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving:		
<div>+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.</div> <div>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.</div>		
<input type="button" value="Wijzig"/>		

rvesten-rsalphen	Cijfer: <input type="text" value="7.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
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).
+De code-stijl is netjes.
-Commentaar is volledig afwezig.

Nette code-stijl, jammer dat de graycode niet gelukt is. De look-and-say reeksen liepen bij mij vast maar zien er qua opbouw goed uit (waarschijnlijk iets met de standaard-bibliotheken).

Wijzig

rtharder

Cijfer: 8.5

Bevestigd: ja Mail

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.

Wijzig

rslagmol

Cijfer: 6.5

Bevestigd: ja Mail

Nakijker: bschuur

Omschrijving:

+De look-and-say reeksen zijn oneindig opgebouwd.
-De Keith-getallen reeksen zijn helaas niet oneindig opgebouwd.
+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 ontstaan negatieve machten. Dit komt omdat je de binnengekomen waarde verlaagd naar de hulpfuncties doorgeeft en deze dan

Wijzig

rpvermeu-jcvrijho

Cijfer: 5.0

Bevestigd: ja Mail

Nakijker: bschuur

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 hoeveelheid 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

Wijzig

rhouweli

Cijfer: 5.5

Bevestigd: ja Mail

Nakijker: bschuur

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.

Wijzig

rhaan	Cijfer: <input type="text" value="8.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving:		
<div>+Je hebt goed gebruik gemaakt van high order functions. +Je hebt goed gebruik gemaakt van lijstcomprehensies. +De code-stijl is netjes. +De Keith getallen zijn oneindig opgebouwd. +De look-and-say reeksen zijn oneindig opgebouwd. -De gray-codering is door de recursieve aanpak wat traag en zorgt voor overflows.</div> <div>In de fromBase en toBase functies schrijf je $n \leq 35$ voor het grondtal maar dit moet $n \leq 36$ zijn, dit zorgt voor fouten bij grondtal 36. Ditzelfde geldt voor de numbers functie. De gCfrom functie in grayCode heeft geen basisgeval en geeft fouten, doordat deze functie recursief gedefinieerd is ontstaan</div>		
<input type="button" value="Wijzig"/>		

rgroot	Cijfer: <input type="text" value="9.0"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: bschuur		
Omschrijving:		
<div>+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).</div> <div>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</div>		
<input type="button" value="Wijzig"/>		

imberg-cwbbonen	Cijfer: <input type="text" value="9.7"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		
<div>[ALGEMEEN] + veel commentaar, nette layout; fijn om naar te kijken. + standaardfuncties gebruikt. [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4]</div>		
<input type="button" value="Wijzig"/>		

jcgoozen	Cijfer: <input type="text" value="5.3"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		
<div>[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</div>		
<input type="button" value="Wijzig"/>		

jcgsmits	Cijfer: <input type="text" value="5.4"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		

<div>[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]</div>		
<div>Wijzig</div>		

jdfeddem	Cijfer: <div>6.3</div>	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + goed gebruik van commentaar, plus mooie layout [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4] + werkt</div>		
<div>Wijzig</div>		

jduijn-iduijn	Cijfer: <div>8.2</div>	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + veel commentaar. + redelijk nette code. [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4]</div>		
<div>Wijzig</div>		

jeceding	Cijfer: <div>6.7</div>	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + nette source file, veel commentaar! [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt.</div>		
<div>Wijzig</div>		

jgageldo	Cijfer: <div>5.6</div>	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + programmacode ziet er verzorgd uit, inclusief commentaar [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4] + werkt</div>		
<div>Wijzig</div>		

jhberg	Cijfer: 6	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: pdwalt		
Omschrijving:		
<div>[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</div>		
<div>Wijzig</div>		

jhorn	Cijfer: 7.4	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: pdwalt		
Omschrijving:		
<div>[ALGEMEEN] + nette becommentarieerde code, soms een beetje spaghetti-achtig [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4] + werkt</div>		
<div>Wijzig</div>		

jjhoozem	Cijfer: 4.9	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: pdwalt		
Omschrijving:		
<div>[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</div>		
<div>Wijzig</div>		

jkoperdr	Cijfer: 8.8	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: pdwalt		
Omschrijving:		
<div>[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]</div>		
<div>Wijzig</div>		

jleersum-anieuwla	Cijfer: 5.5	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: pdwalt		
Omschrijving:		

<div>[ALGEMEEN] + leesbare code met commentaar. goed gebruik van standaardfuncties. [OPGAVE 1] +werkt [OPGAVE 2] +werkt [OPGAVE 3] +werkt [OPGAVE 4] +werkt</div>		
<div>Wijzig</div>		

jmlinsse	Cijfer: 6.3	Bevestigd: ja Mail
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + nette code, erg veel commentaar. + we worden blij van standaardfuncties. [OPGAVE 1] + werkt [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4]</div>		
<div>Wijzig</div>		

jmulder	Cijfer: 7.6	Bevestigd: ja Mail
Nakijker: pdwalt		
<div>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]</div>		
<div>Wijzig</div>		

jtkman-echgbon	Cijfer: 6.3	Bevestigd: ja Mail
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + code maakt veel gebruik van standaardfuncties en is goed becommentarieerd. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt.</div>		
<div>Wijzig</div>		

jwind	Cijfer: 5.2	Bevestigd: ja Mail
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + programmeerstijl is netjes: korte bondige functies zonder overbodige herdefinities van standaardfuncties. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4]</div>		
<div>Wijzig</div>		

kfaro	Cijfer: <input type="text" value="5.5"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		
<div>[ALGEMEEN] + veel commentaar, leesbare code, + veel gebruik van standaardfuncties [OPGAVE 1] + werkt. [OPGAVE 2] + werkt [OPGAVE 3] + werkt [OPGAVE 4]</div>		
<input type="button" value="Wijzig"/>		

kjdvoors-apol	Cijfer: <input type="text" value="4.1"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		
<div>[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]</div>		
<input type="button" value="Wijzig"/>		

lblhartm	Cijfer: <input type="text" value="1.2"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		
<div>[ALGEMEEN] * nou, karig. + hetgeen aanwezig was werkte wel tenminste. [OPGAVE 1] + werkt. [OPGAVE 2] - ontbreekt. [OPGAVE 3] + werkt. [OPGAVE 4]</div>		
<input type="button" value="Wijzig"/>		

ldspbroe-jfklein	Cijfer: <input type="text" value="7.6"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		
<div>[ALGEMEEN] + programmeerstijl is redelijk netjes, leesbare code. + commentaar. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4]</div>		
<input type="button" value="Wijzig"/>		

lrwester	Cijfer: <input type="text" value="4.2"/>	Bevestigd: <input type="text" value="ja"/> <input type="button" value="Mail"/>
Nakijker: pdwalt		
Omschrijving:		

<div>[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]</div>		
<div>Wijzig</div>		

Isstoel	Cijfer: 9.3	Bevestigd: ja Mail
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + mooie code met veel commentaar dat bovendien leesbaar is. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4] + werkt.</div>		
<div>Wijzig</div>		

Itbinsbe	Cijfer: 8	Bevestigd: ja Mail
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + nette code met veel commentaar. prima layout. + gebruik van standaardfuncties. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4]</div>		
<div>Wijzig</div>		

Iwgraaff	Cijfer: 6.7	Bevestigd: ja Mail
Nakijker: pdwalt		
<div>Omschrijving: [ALGEMEEN] + code handig afgebakend in secties; standaardfuncties worden gebruikt (foldl, yay). + commentaar. [OPGAVE 1] + werkt. [OPGAVE 2] + werkt. [OPGAVE 3] + werkt. [OPGAVE 4]</div>		
<div>Wijzig</div>		

elrenkem-ogrottie	Cijfer: 9.8	Bevestigd: ja Mail
Nakijker: rkoot		
<div>Omschrijving: [ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd. [OPGAVE 1] Prima. [OPGAVE 2]</div>		
<div>Wijzig</div>		

bdoren-mjspoor	Cijfer: 7.6	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Voldoende commentaar.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>		
<div>Wijzig</div>		

bgreeven-jsteenbe	Cijfer: 6.9	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Voldoende commentaar.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>		
<div>Wijzig</div>		

bjliefer-ptpkokke	Cijfer: 9.8	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentarieerd.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>		
<div>Wijzig</div>		

bspaans	Cijfer: 8.5	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentarieerd.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>		
<div>Wijzig</div>		

btdijk	Cijfer: 8.5	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		

<div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Voldoende commentaar.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>		<div></div>
		<div>Wijzig</div>

<div>cjlblom</div>	<div>Cijfer: 8.5</div>	<div>Bevestigd: ja</div>	<div>Mail</div>
<div>Nakijker: rkoot</div>			
<div>Omschrijving:</div> <div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>			
<div>Wijzig</div>			

<div>ddtoniss-dsgroote</div>	<div>Cijfer: 7.2</div>	<div>Bevestigd: ja</div>	<div>Mail</div>
<div>Nakijker: rkoot</div>			
<div>Omschrijving:</div> <div>[ALGEMEEN] Code is redelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>			
<div>Wijzig</div>			

<div>dekuppev</div>	<div>Cijfer: 8.5</div>	<div>Bevestigd: ja</div>	<div>Mail</div>
<div>Nakijker: rkoot</div>			
<div>Omschrijving:</div> <div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>			
<div>Wijzig</div>			

<div>dgerritz</div>	<div>Cijfer: 7.9</div>	<div>Bevestigd: ja</div>	<div>Mail</div>
<div>Nakijker: rkoot</div>			
<div>Omschrijving:</div> <div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>			
<div>Wijzig</div>			

dpboot	Cijfer: 7.3	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Code is duidelijk leesbaar. Weinig gebruik gemaakt van standaardfuncties. Code is goed becommentarieerd.</div><div>[OPGAVE 1] Dit had efficiënter geïmplementeerd kunnen worden zonder gebruik te maken van machtsverheffen.</div><div>[OPGAVE 2] Prima.</div></div>		
Wijzig		

dwinkel	Cijfer: 6.8	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is voldoende becommentarieerd.</div><div>[OPGAVE 1] Dit had efficiënter geïmplementeerd kunnen worden zonder gebruik te maken van machtsverheffen.</div><div>[OPGAVE 2] Prima.</div></div>		
Wijzig		

affboth-lveerman	Cijfer: 9.6	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Commentaar is spaarzaam gebruikt.</div><div>[OPGAVE 1] fromBase: Een oplossing zonder gebruik te maken van machtsverheffen was efficiënter geweest.</div><div>[OPGAVE 2] Prima.</div></div>		
Wijzig		

ergallo	Cijfer: 9.5	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.</div><div>[OPGAVE 1] Prima.</div><div>[OPGAVE 2]</div></div>		
Wijzig		

ewjmulde	Cijfer: 10.0	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: rkoot		
Omschrijving:		

gdijkstr-rjhensin	Cijfer: 8.8	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is matig becommentarieerd.</div><div>[OPGAVE 1] Prima.</div><div>[OPGAVE 2] Prima.</div></div>		
<div>Wijzig</div>		

gloupias	Cijfer: 8.8	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is voldoende becommentarieerd.</div><div>[OPGAVE 1] Dit had efficiënter geïmplementeerd kunnen worden zonder gebruik te maken van machtsverheffen.</div><div>[OPGAVE 2] Prima.</div></div>		
<div>Wijzig</div>		

hjkuijk	Cijfer: 7.1	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Code is duidelijk leesbaar. Weinig gebruik gemaak van standaardfuncties. Code is onvoldoende becommentarieerd.</div><div>[OPGAVE 1] Dit had efficiënter gedaan kunnen worden zonder gebruik te maken van machtsverheffen.</div><div>[OPGAVE 2] Prima.</div></div>		
<div>Wijzig</div>		

hlversto	Cijfer: 8.2	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: rkoot		
Omschrijving:		
<div><div>[ALGEMEEN] Code is duidelijk leesbaar. Niet veel gebruik gemaak van standaard- en hulpfuncties. Code is onvoldoende becommentarieerd.</div><div>[OPGAVE 1] Prima.</div><div>[OPGAVE 2] Prima.</div></div>		
<div>Wijzig</div>		

hmpaasse	Cijfer: 7.8	Bevestigd: <input type="button" value="ja"/> <input type="button" value="Mail"/>
Nakijker: rkoot		
Omschrijving:		

<div>[ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is onvoldoende becommentarieerd.</div> <div>[OPGAVE 1] Prima.</div> <div>[OPGAVE 2] Prima.</div>		<div></div>
		<div>Wijzig</div>

reahoef-irrencke	Cijfer: 8.7	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: svisser		
<div>Omschrijving:</div> <div>[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.</div> <div>[OPGAVE 1]</div>		
		<div>Wijzig</div>

rawagenm	Cijfer: 8.8	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: svisser		
<div>Omschrijving:</div> <div>[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:</div>		
		<div>Wijzig</div>

ravries-ccmrooij	Cijfer: 7.6	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: svisser		
<div>Omschrijving:</div> <div>[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</div>		
		<div>Wijzig</div>

raspauwe	Cijfer: 7.7	Bevestigd: <div>ja</div> <div>Mail</div>
Nakijker: svisser		
<div>Omschrijving:</div> <div>[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.</div>		
		<div>Wijzig</div>

pqgroot	Cijfer: 9.3	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svisser		
Omschrijving:		
<div>[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</div>		
<div>Wijzig</div>		

pmspek	Cijfer: 10	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svisser		
Omschrijving:		
<div>[ALGEMEEN] Weinig aan toe te voegen: een prima practicum. Er wordt goed gebruik gemaakt van standaardfuncties en accumulerende parameters om de efficiëntie te verbeteren. De functies leveren ook het juiste resultaat op. [OPGAVE 1] OK. [OPGAVE 2]</div>		
<div>Wijzig</div>		

pjwjanse-jjvisser	Cijfer: 7.7	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svisser		
Omschrijving:		
<div>[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.</div>		
<div>Wijzig</div>		

pdstaats-cjplatte	Cijfer: 6.0	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svisser		
Omschrijving:		
<div>[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</div>		
<div>Wijzig</div>		

ompennin	Cijfer: 8.7	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svisser		
Omschrijving:		

<div>[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 = ...</div>		<div></div>
		<div>Wijzig</div>

nroumimp	Cijfer: <div>8.7</div>	Bevestigd: <div>ja</div> <div>Mail</div>	
Nakijker: svisser			
<div>Omschrijving: [ALGEMEEN] De code is bij de meeste opgaven kort wat een indicatie kan zijn dat er een goede aanpak is gekozen (zoals het gebruik van een foldfunctie waar dat mogelijk is). Het werkt ook naar behoren. [OPGAVE 1] OK. [OPGAVE 2] OK.</div>			
			<div>Wijzig</div>

njvlamin	Cijfer: <div>4.4</div>	Bevestigd: <div>ja</div> <div>Mail</div>	
Nakijker: svisser			
<div>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:</div>			
			<div>Wijzig</div>

mzwan-jgeeden	Cijfer: <div>5.2</div>	Bevestigd: <div>ja</div> <div>Mail</div>	
Nakijker: svisser			
<div>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 hoeft je ze niet letterlijk over te nemen. [OPGAVE 1]</div>			
			<div>Wijzig</div>

mvensela-merboxel	Cijfer: <div>8.1</div>	Bevestigd: <div>ja</div> <div>Mail</div>	
Nakijker: svisser			
<div>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 :-)</div>			
			<div>Wijzig</div>

mtibboel-sabitter	Cijfer: 7.3	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svissser		
Omschrijving:		
<div>[ALGEMEEN] De code bestaat uit korte functies doordat er hulpfuncties worden gebruikt waar nodig. Over het algemeen een goed practicum maar er zijn enkele dingen die niet goed worden afgehandeld (bijvoorbeeld door de gekozen aanpak, bij opgave 9, of door randgevallen, zoals bij opgave 2).</div> <div>[OPGAVE 1] OK.</div>		
Wijzig		

mtduysen	Cijfer: 2.6	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svissser		
Omschrijving:		
<div>[ALGEMEEN] De eerste opgaven zijn gemaakt en werken ook maar het leeuwendeel van het practicum ontbreekt.</div> <div>[OPGAVE 1] OK.</div> <div>[OPGAVE 2] OK.</div>		
Wijzig		

mrvaarti-rwerken	Cijfer: 7.8	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svissser		
Omschrijving:		
<div>[ALGEMEEN] De meeste opgaven werken naar behoren alleen bij opgave 7 en 8 ontbreekt er enige functionaliteit (probeer de opgave goed te bestuderen).</div> <div>Er worden lambda-expressies gebruikt waar dat mogelijk is en ook de standaardfuncties worden benut waardoor veel functies uit slechts 1 regel bestaan.</div> <div>[OPGAVE 1]</div>		
Wijzig		

mmontvai-beck	Cijfer: 7.0	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svissser		
Omschrijving:		
<div>[ALGEMEEN] Over het algemeen een goed practicum; de functies zijn kort door het gebruik van standaardfuncties en hulpfuncties.</div> <div>De Gray-opgave ontbreekt en er zijn enkele situaties waarbij een verkeerd resultaat wordt teruggegeven maar verder werkt het. De code maakt wel veel gebruik van if then else constructies terwijl het soms ook 'functioneler' kan.</div> <div>[OPGAVE 1]</div>		
Wijzig		

mlmbroer-jmwbrete	Cijfer: 9.1	Bevestigd: <input type="radio"/> ja <input type="radio"/> Mail
Nakijker: svissser		
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

Wijzig

mkroese

Cijfer: 9.2

Bevestigd: ja Mail

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]

Wijzig

mgrimme

Cijfer: 9.5

Bevestigd: ja Mail

Nakijker: svisser

Omschrijving:

[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.

Wijzig

mfrancke-jfdijks

Cijfer: 7.7

Bevestigd: ja Mail

Nakijker: svisser

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).

Wijzig

mbarendr

Cijfer: 5.3

Bevestigd: ja Mail

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.

Wijzig

maooster	Cijfer: 9.2	Bevestigd: ja Mail
Nakijker: svisser		
Omschrijving:		
<div>[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</div>		
Wijzig		

mahashi-mjhobbel	Cijfer: 8.2	Bevestigd: ja Mail
Nakijker: svisser		
Omschrijving:		
<div>[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.</div>		
Wijzig		

sttimmer	Nakijker: bschuur	Cijfer: 10
Omschrijving:		
<div>+De gray-codering is efficiënt 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.</div>		
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.		
flidenis	Nakijker: rkoot	Cijfer: 10
Omschrijving:		
<div>[ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaak van standaard- en hulpfuncties. Code is goed becommentariëerd.</div>		
[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] Prima.		
[OPGAVE 10] Prima.		
pmspek	Nakijker: svisser	Cijfer: 10
Omschrijving: [ALGEMEEN] Weinig aan toe te voegen: een prima practicum. Er wordt goed gebruik gemaakt van standaardfuncties en accumulerende parameters om de efficiëntie te verbeteren. De functies leveren ook het juiste resultaat op.		
[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	Nakijker: rkoot	Cijfer: 10.0
Omschrijving: [ALGEMEEN] Uitstekend! Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.		
[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] 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.		
mtduysen	Nakijker: svisser	Cijfer: 2.6
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 namelijk ook letters worden meegegeven (zie ook het voorbeeld). [OPGAVE 6] Niet gemaakt. [OPGAVE 7] Niet gemaakt. [OPGAVE 8] Niet gemaakt. [OPGAVE 9] Niet gemaakt.		

[OPGAVE 10] Niet gemaakt.		
talles	Nakijker: bschuur	Cijfer: 4.0
<p>Omschrijving: +De code-stijl is netjes. +Het commentaar is goed en effectief.</p> <p>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.</p>		
kjdvoors-apol	Nakijker: pdwalt	Cijfer: 4.1
<p>Omschrijving: [ALGEMEEN] + code is wel bondig, MAAR - gebruikt a.u.b. nooit meer namen als 'hulpfunctie101' en aanverwanten!</p> <p>[OPGAVE 1] + werkt.</p> <p>[OPGAVE 2] + werkt.</p> <p>[OPGAVE 3] + werkt.</p> <p>[OPGAVE 4] + werkt.</p> <p>[OPGAVE 5] + werkt.</p> <p>[OPGAVE 6] + werkt.</p> <p>[OPGAVE 7] - off-by-one error: *Main> G.numbers 31 ["rvhokcn"] [["rvhokcn",-1790769933]] zou geen resultaten moeten geven. (v==31 in base 32)</p> <p>[OPGAVE 8] - ontbreekt.</p> <p>[OPGAVE 9] - ontbreekt.</p> <p>[OPGAVE 10] + werkt. - heel erg niet oneindig.</p>		
lrwester	Nakijker: pdwalt	Cijfer: 4.2
<p>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.</p> <p>[OPGAVE 1] + werkt.</p> <p>[OPGAVE 2] + werkt.</p> <p>[OPGAVE 3] + werkt.</p> <p>[OPGAVE 4] + werkt.</p> <p>[OPGAVE 5] + werkt.</p> <p>[OPGAVE 6] + werkt.</p> <p>[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.</p> <p>[OPGAVE 8] - doet alsof base 2 gevraagd wordt (negeert karakters anders dan [0,1]) + niet heel inefficiënt</p> <p>[OPGAVE 9] - gaat nogal mis in het volgende voorbeeld: *Main> G.lookAndSay 111 ["9","19","1119","3119","132119", ...]</p> <p>[OPGAVE 10] - gaat vanaf 7e element mis. Die moet namelijk 197 zijn, niet 290. + wel mooie lijst (oneindig)</p>		
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 1212121212

[9,5,3,0,3,6,9,2,4]

[OPGAVE 3]

+ werkt

[OPGAVE 4]

+ werkt

[OPGAVE 5]

+ werkt

[OPGAVE 6]

- heeft last van hetzelfde als toDec

[OPGAVE 7]

- foutje: (off-by-one)

numbers 24 ["co"]

[("co",312)]

moet de lege lijst teruggeven... [OPGAVE 8] - bij encoderen van grayCode wordt base als "lengte" geïnterpreteerd? en decoderen geeft elke keer 2^{base} ... [OPGAVE 9] - lookAndSay 0 gaat al mis... geeft ["0", "", "" ...] [OPGAVE 10] + 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.		
sagieske	Nakijker: bschuur	Cijfer: 5.0
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 hoeveelheid 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 hoeft 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 $xs \geq 0 \ \&\& \ xs \leq 10$). 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 controleren 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.

jegoosen	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 ["judjzcejrlagrq","riv"] eens (z mag er niet in voorkomen).
[OPGAVE 8]
- fromGray ontbreekt
[OPGAVE 9]
- ontbreekt
[OPGAVE 10]
+ werkt
+ oneindige lijst

mbarendr	Nakijker: svisser	Cijfer: 5.3
----------	-------------------	-------------

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 l size = csum : genKeithGetallen (l ++ [csum]) size
where csum = sum (take size (reverse l))
```

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", ...]

[OPGAVE 10]

+ oneindige lijst geconstrueerd

+ werkt

fleroi	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 aangeroepen 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
["0","10","1110","3110","132110","1113122110","311311222110","12101713151716171918","111814101711171926","211613182932"]
moet zijn:
["0","10","1110","3110","132110","1113122110","311311222110","13211321322110","1113122113121113222110","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		
--	--	--

fbijlsm	Nakijker: rkoot	Cijfer: 5.7
Omschrijving: [ALGEMEEN] Code is slorig. Weing gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd. [OPGAVE 1] Dit had efficiënter geïmplementeerd 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 efficiënt 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 tuple, 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
<p>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</p>		
jdfeddem	Nakijker: pdwalt	Cijfer: 6.3
<p>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" 19 opeenvolgende getallen moeten 1 plaats verschillen: *Getallen> (snd\$grayCode 2) 110012 "10011011011010111" *Getallen> (snd\$grayCode 2) 110011 "10011011011010010" ^ ^ + verder wel redelijk efficient geïmplementeerd. [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 inefficiënt.</p>		

jtkman-echgbon	Nakijker: pdwalt	Cijfer: 6.3
<p>Omschrijving:</p> <p>[ALGEMEEN]</p> <p>+ code maakt veel gebruik van standaardfuncties en is goed becommentarieerd.</p> <p>[OPGAVE 1]</p> <p>+ werkt.</p> <p>[OPGAVE 2]</p> <p>+ werkt.</p> <p>[OPGAVE 3]</p> <p>+ werkt.</p> <p>[OPGAVE 4]</p> <p>+ werkt.</p> <p>[OPGAVE 5]</p> <p>+ werkt.</p> <p>[OPGAVE 6]</p> <p>+ werkt.</p> <p>[OPGAVE 7]</p> <p>- hoe is dit correct?</p> <p>*Main> G.numbers 24 ["jz"]</p> <p>[]</p> <p>makkelijker is waarschijnlijk gewoon wegstrepen wat niet mag volgens base-alfabet.</p> <p>[OPGAVE 8]</p> <p>- niet zijn eigen inverse!</p> <p>- ook erg inefficiënte implementatie</p> <p>[OPGAVE 9]</p> <p>- jammer, dit:</p> <p>*Main> G.lookAndSay 9</p> <p>[""]</p> <p>in jullie commentaar staat dat het anders niet klopt, nou je kan met 9 gewoon</p> <p>["19","1119" ..] maken...</p> <p>+ verder werkt het wel.</p> <p>[OPGAVE 10]</p> <p>+ werkt.</p> <p>+ mooie lijst-oplossing.</p>		
rslagmol	Nakijker: bschuur	Cijfer: 6.5
<p>Omschrijving:</p> <p>+De look-and-say reeksen zijn oneindig opgebouwd.</p> <p>-De Keith-getallen reeksen zijn helaas niet oneindig opgebouwd.</p> <p>+Je hebt goed gebruik gemaakt van high order functions. Je had hier nog meer gebruik van kunnen maken bij de eerste opgaven.</p> <p>-Je code is soms wat rommelig, bijvoorbeeld een functie ‘rekenen a z y x’ is compleet onduidelijk wat het doet.</p> <p>-Er mocht her en der wat meer commentaar bij. Ik mis het commentaar bij juist de moeilijke gedeeltes.</p> <p>Bij de toBin en toBase functies verslikt hij zich als 0 wordt ingegeven, dan ontstaan 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.</p>		
jeceding	Nakijker: pdwalt	Cijfer: 6.7
<p>Omschrijving:</p> <p>[ALGEMEEN]</p> <p>+ nette source file, veel commentaar!</p> <p>[OPGAVE 1]</p> <p>+ werkt.</p> <p>[OPGAVE 2]</p> <p>+ werkt.</p> <p>[OPGAVE 3]</p> <p>+ werkt.</p> <p>[OPGAVE 4]</p> <p>+ werkt.</p> <p>[OPGAVE 5]</p> <p>+ werkt.</p> <p>[OPGAVE 6]</p> <p>+ werkt.</p> <p>[OPGAVE 7]</p> <p>+ werkt.</p> <p>[OPGAVE 8]</p> <p>+ last van het lege String in plaats van "0" probleem... verder bijna ok.</p> <p>wel goed over nagedacht.</p> <p>[OPGAVE 9]</p> <p>- kop klopt niet helemaal -> lookAndSay 0 = ["" , "10" ..] moet met "0" beginnen.</p> <p>maar verder bijna juist</p> <p>+ oneindige lijst</p> <p>[OPGAVE 10]</p> <p>+ werkt op zich wel, alleen erg langzaam. Inefficiënte implementatie.</p> <p>+ oneindige lijst</p>		

lwgraaff	Nakijker: pdwalt	Cijfer: 6.7
<p>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 inefficiënt [OPGAVE 9] + werkt. [OPGAVE 10] + werkt. + mooie oneindige lijst.</p>		
dwinkel	Nakijker: rkoot	Cijfer: 6.8
<p>Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is voldoende becommentarieerd. [OPGAVE 1] Dit had efficiënter geïmplementeerd 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, maar is te inefficiënt om meer dan een handvol look-and-say te genereren. [OPGAVE 10] Werkt, maar is te inefficiënt om meer dan een handvol Keith-getallen te genereren.</p>		
bgreeven-jsteenbe	Nakijker: rkoot	Cijfer: 6.9
<p>Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties.</p>		

Voldoende commentaar.

[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
--------	-------------------	-------------

Omschrijving:

-Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie).

+De Keith getallen zijn oneindig opgebouwd.

+De look-and-say reeksen zijn oneindig opgebouwd.

-Commentaar is volledig afwezig.

Alhoewel al je to* functies lijken te kloppen kunnen ze niet goed omgaan met grote getallen, ik heb niet kunnen uitvissen waar hem dat in zit(waarschijnlijk beperkingen van int). Je getCharacter functie in toBase bevat een onvolledig patroon. Je numbers functie laat ook alle letters door die gelijk zijn aan het grondtal, dit moeten echter alle letters zijn die eentje minder zijn dan het grondtal.

mmontvai-beck	Nakijker: svisser	Cijfer: 7.0
---------------	-------------------	-------------

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 het.

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.

[OPGAVE 5]

OK.

[OPGAVE 6]

OK.

[OPGAVE 7]

De functie possible laat ook toe dat de letter die net buiten de geldige tekenreeks valt, gebruikt mag worden. Bijvoorbeeld: de letter 'g' mag bij grondtal 16 ook gebruikt worden terwijl alleen de getallen en 'a' t/m 'f' toegestaan zijn.

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 gemaakt van standaardfuncties.

Code is onvoldoende becommentarieerd.

[OPGAVE 1]

Dit had efficiënter 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 gemaakt van standaard- en hulpfuncties.

Code is goed becommentarieerd.

[OPGAVE 1]

Prima.

[OPGAVE 2]

Prima.

[OPGAVE 3]

fromBin: Een oplossing zonder gebruik te maken van machtsverheffen was efficiënter geweest.

[OPGAVE 4]
Prima

[OPGAVE 5]
parsePowers: Hier hadden jullie gebruik kunnen maken van de standaardfunctie "ord".

[OPGAVE 6]
Prima.

[OPGAVE 7]
De functie is incorrect:
* Hij had wel degelijk moeten werken op getallen met een grondgetal kleiner dan 10, ook al produceert dit altijd een lege lijst.
* Er zit een off-by-one error in jullie implementatie (numbers 16 ["f", "g", "h"] levert [("f",15),("g",16)] op in plaats van [("f",15)]).

[OPGAVE 8]
Jullie implmentatie komt door geen enkele van onze testcases heen 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 gemaakt van standaardfuncties.
Code is goed becommentarieerd.

[OPGAVE 1]
Dit had efficiënter geïmplementeerd 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 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 worden gebruikt waar nodig. Over het algemeen een goed practicum maar er 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, fromBase base x)
isGeldig w = ...
```

Een manier om de functie isGeldig te schrijven kan door een alfabet 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.

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. lookAndSay 789. (waardoor automatisch testen faalt) + klopt wel voor lagere beginwaarden. [OPGAVE 10] + werkt		
wokatz	Nakijker: bschuur	Cijfer: 7.5
Omschrijving: +De look-and-say reeksen zijn oneindig opgebouwd maar lopen tegen de beperkingen van een integer aan. +Je hebt goed gebruik gemaakt van high order functions. +De code-stijl is netjes. +Het commentaar is goed en effectief. Er zitten wat subtiele foutje in je werk. Zo heb je bij toBin het basisgeval 0 niet afgevangen wat het programma in een oneindige loop gooit. Verder klopt de numbers functie niet helemaal, de getallen met dezelfde waarde als het grondtal worden nog steeds als kloppend gezien(dit moet 1 minder zijn).		
sjveldhu	Nakijker: bschuur	Cijfer: 7.5
Omschrijving: +De look-and-say reeksen zijn oneindig opgebouwd, maar gaat op een gegeven moment de fout in (Integer inplaats van Int gebruiken?). +De Keith getallen zijn oneindig opgebouwd. -Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie). Zeker bij de eerste opgaven. +De code-stijl is netjes. +Het commentaar is goed en effectief.		
rvesten-rsalphen	Nakijker: bschuur	Cijfer: 7.5
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. -Commentaar is volledig afwezig. Nette code-stijl, jammer dat de graycode niet gelukt is. De look-and-say reeksen liepen bij mij vast maar zien er qua opbouw goed uit (waarschijnlijk iets met de standaard-bibliotheken).		
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 5] + werkt. [OPGAVE 6] + werkt. [OPGAVE 7] - helaas off-by-one error: *Getallen> numbers 34 ["pry"] [("pry",29852)] terwijl y == 34 in base 35 [OPGAVE 8] - doet het alleen alsof base 2 gevraagd wordt (opmerking gezien, enigszins		

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

bdoren-mjspoor	Nakijker: rkoot	Cijfer: 7.6
----------------	-----------------	-------------

Omschrijving:
[ALGEMEEN]
Code is duidelijk leesbaar.
Handig gebruik gemaakt 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	Nakijker: rkoot	Cijfer: 7.6
----------	-----------------	-------------

Omschrijving:

[ALGEMEEN]

Code is duidelijk leesbaar.

Weinig gebruik gemaakt van standaardfuncties.

Code is goed becommentarieerd.

[OPGAVE 1]

Dit had efficiënter geïmplementeerd 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.

ravries-ccmrooij

Nakijker: svisser

Cijfer: 7.6

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 $\leq 65 + \text{grond} - 10$ doen i.p.v. $< 65 + \text{grond} - 10$ 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-jfdijks	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.

raspauwe	Nakijker: svisser	Cijfer: 7.7
----------	-------------------	-------------

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 hoeft 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 $\leq n$ doet i.p.v. $< n$ wanneer je met het grondtal vergelijkt.

[OPGAVE 8]

Wel een begin mee gemaakt maar niet voltooid.

[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]

OK.

pjwjanse-jjvisser

Nakijker: svisser

Cijfer: 7.7

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 tuple).

[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..]
```

hmpaasse

Nakijker: rkoot

Cijfer: 7.8

Omschrijving:

[ALGEMEEN]

Code is duidelijk leesbaar.

Handig gebruik gemaakt van standaard- en hulpfuncties.

Code is onvoldoende becommentarieerd.

[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, efficiënte implementatie.

[OPGAVE 9]
Werkt niet.

[OPGAVE 10]
Prima.

mrvaarti-rwerken

Nakijker: svisser

Cijfer: 7.8

<p>Omschrijving:</p> <p>[ALGEMEEN]</p> <p>De meeste opgaven werken naar behoren alleen bij opgave 7 en 8 ontbreekt er enige functionaliteit (probeer de opgave goed te bestuderen).</p> <p>Er worden lambda-expressies gebruikt waar dat mogelijk is en ook de standaardfuncties worden benut waardoor veel functies uit slechts 1 regel bestaan.</p> <p>[OPGAVE 1]</p> <p>OK.</p> <p>[OPGAVE 2]</p> <p>OK.</p> <p>[OPGAVE 3]</p> <p>OK.</p> <p>[OPGAVE 4]</p> <p>OK.</p> <p>[OPGAVE 5]</p> <p>OK.</p> <p>[OPGAVE 6]</p> <p>OK.</p> <p>[OPGAVE 7]</p> <p>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).</p> <p>[OPGAVE 8]</p> <p>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.</p> <p>Zo levert toGray 10 7 de string "4" terwijl toGray 10 8 de string "12" oplevert waardoor er meer dan 1 teken verandert.</p> <p>[OPGAVE 9]</p> <p>OK.</p> <p>[OPGAVE 10]</p> <p>OK.</p>		
dgerritz	Nakijker: rkoot	Cijfer: 7.9
<p>Omschrijving:</p> <p>[ALGEMEEN]</p> <p>Code is duidelijk leesbaar.</p> <p>Handig gebruik gemaakt van standaard- en hulpfuncties.</p> <p>Code is goed becommentarieerd.</p> <p>[OPGAVE 1]</p> <p>Prima.</p> <p>[OPGAVE 2]</p> <p>Prima.</p> <p>[OPGAVE 3]</p> <p>Dit had efficiënter geïmplementeerd kunnen worden zonder gebruik te maken van machtsverheffen.</p> <p>[OPGAVE 4]</p> <p>Prima.</p> <p>[OPGAVE 5]</p> <p>Prima.</p> <p>[OPGAVE 6]</p> <p>Prima.</p> <p>[OPGAVE 7]</p> <p>Prima.</p>		

[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 6]

+ werkt.

[OPGAVE 7]

+ werkt.

[OPGAVE 8]

- ontbreekt.

[OPGAVE 9]

+ werkt.

[OPGAVE 10]

+ werkt.

+ mooie oneindige lijst.

sjavis	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.v. recursie, lijstcomprehensie).

+Het commentaar is goed en effectief.

Je zou meer gebruik kunnen maken van patroonherkenning(bijvoorbeeld op lijsten, dan hoef je niet steeds head en tail aan te roepen). Ook is het gebruik van meer hogere orde functies aan te raden, deze beide kunnen de grootte van je code flink inperken en de leesbaarheid vergroten.

tmssoetho	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 minder goed weer wat er bedoeld wordt.

[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 $y + 48$ 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 tuple te produceren).

[OPGAVE 8]
De functie grayToInt werkt niet voor "0" maar verder werkt het wel. De waarde van y is dan gelijk aan $a^0 + 0 = 1$ 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

Nakijker: pdwalt

Cijfer: 8.2

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 ["zzzzzzzzzzzz"]
```

```
*** 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 gemaakt van standaard- en hulpfuncties.

Code is onvoldoende becommentarieerd.

[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 isKeith x = elem x (keithReeks x)
```

rtharder	Nakijker: bschuur	Cijfer: 8.5
-----------------	-------------------	-------------

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

en der kunnen kijken of er een kortere oplossing was voor de problemen.		
tromberg-rjanssen	Nakijker: bschuur	Cijfer: 8.5
<p>Omschrijving:</p> <p>-De gray-codering is door de recursieve aanpak wat traag en zorgt voor overflows.</p> <p>+De look-and-say reeksen zijn oneindig opgebouwd.</p> <p>+De Keith getallen zijn oneindig opgebouwd.</p> <p>+Je hebt goed gebruik gemaakt van high order functions, vooral in look-and say, je had dit nog meer kunnen toepassen.</p> <p>+De code-stijl is netjes.</p> <p>Doordat de graycode functie een oneindige lijst opbouwt wordt deze te traag, je had hier beter een directe formula kunnen proberen te vinden. Je numbers functie filtert foute woorden er niet uit, had je de opdracht goed begrepen?</p>		
whustinx	Nakijker: bschuur	Cijfer: 8.5
<p>Omschrijving:</p> <p>+De gray-codering is efficiënt opgelost maar geeft helaas ‘division by zero’ fouten. Hier moet misschien ergens een randgeval afgevangen worden.</p> <p>+De Keith getallen zijn oneindig opgebouwd.</p> <p>+De look-and-say reeksen zijn oneindig opgebouwd.</p> <p>+Je hebt goed gebruik gemaakt van high order functions.</p> <p>+De code-stijl is netjes, maar vooral de naamgeving is naar het einde toe cryptisch.</p> <p>-Commentaar is volledig afwezig.</p> <p>In je numbers functie zit een subtiele fout: een getal met dezelfde waarde als het grondtal wordt geaccepteerd, dit hoort niet. Als je cryptische naamgeving voor functies gebruikt is commentaar vereist(zie Keith).</p>		
rhaan	Nakijker: bschuur	Cijfer: 8.5
<p>Omschrijving:</p> <p>+Je hebt goed gebruik gemaakt van high order functions.</p> <p>+Je hebt goed gebruik gemaakt van lijstcomprehensies.</p> <p>+De code-stijl is netjes.</p> <p>+De Keith getallen zijn oneindig opgebouwd.</p> <p>+De look-and-say reeksen zijn oneindig opgebouwd.</p> <p>-De gray-codering is door de recursieve aanpak wat traag en zorgt voor overflows.</p> <p>In de fromBase en toBase functies schrijf je $n \leq 35$ voor het grondtal maar dit moet $n \leq 36$ zijn, dit zorgt voor fouten bij grondtal 36. Ditzelfde geldt voor de numbers functie. De gCfrom functie in grayCode heeft geen basisgeval en geeft fouten, doordat deze functie recursief gedefinieerd is ontstaan stack overflows bij grote getallen en is de functie traag.</p>		
btdijk	Nakijker: rkoot	Cijfer: 8.5
<p>Omschrijving:</p> <p>[ALGEMEEN]</p> <p>Code is duidelijk leesbaar.</p> <p>Handig gebruik gemaakt van standaard- en hulpfuncties.</p> <p>Voldoende commentaar.</p> <p>[OPGAVE 1]</p> <p>Prima.</p> <p>[OPGAVE 2]</p> <p>Prima.</p> <p>[OPGAVE 3]</p> <p>Prima.</p> <p>[OPGAVE 4]</p> <p>Prima.</p> <p>[OPGAVE 5]</p> <p>Prima.</p> <p>[OPGAVE 6]</p> <p>Prima.</p> <p>[OPGAVE 7]</p> <p>Prima.</p> <p>[OPGAVE 8]</p> <p>Werkt, maar inefficiënt geïmplementeerd.</p> <p>[OPGAVE 9]</p> <p>Gaat mis voor het randgeval 0.</p> <p>Te inefficiënt om de eerste 15 te genereren.</p> <p>[OPGAVE 10]</p> <p>Prima.</p>		

<p>Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.</p> <p>[OPGAVE 1] Prima.</p> <p>[OPGAVE 2] Prima.</p> <p>[OPGAVE 3] Prima.</p> <p>[OPGAVE 4] Prima.</p> <p>[OPGAVE 5] Prima.</p> <p>[OPGAVE 6] Prima.</p> <p>[OPGAVE 7] Controleert niet of het woord wel een geldig getal is in de opgegeven basis (bijvoorbeeld: numbers 17 ["z"] had de lege lijst op moeten leveren, niet [{"z",35}])).</p> <p>[OPGAVE 8] Niet inverteerbaar voor bijvoorbeeld: *Getallen> snd (grayCode 11) 121 "1a0" *Getallen> fst (grayCode 11) "1a0" 122</p> <p>[OPGAVE 9] Werkt niet voor startwaardes groter dan 9.</p> <p>[OPGAVE 10] Prima</p>		
---	--	--

<p>Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.</p> <p>[OPGAVE 1] Prima.</p> <p>[OPGAVE 2] Prima.</p> <p>[OPGAVE 3] Prima.</p> <p>[OPGAVE 4] Prima.</p> <p>[OPGAVE 5] Prima.</p> <p>[OPGAVE 6] Prima.</p> <p>[OPGAVE 7] Werkt niet altijd correct, bijvoorbeeld op het geval numbers 27 ["ed","ylfey","iroojpi","otqswf"].</p> <p>[OPGAVE 8] Werkt, maar is erg inefficiënt.</p> <p>[OPGAVE 9] Werkt niet voor startwaardes groter dan 9.</p>		
---	--	--

[OPGAVE 10] Prima.		
bspaans	Nakijker: rkoot	Cijfer: 8.5
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.		
[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 inefficiënt om de eerste 15 te genereren.		
[OPGAVE 10] Te inefficiënt 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 zijn dat er een goede aanpak is gekozen (zoals het gebruik van een foldfunctie waar dat mogelijk is). Het werkt ook naar behoren.		
[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, zoals filter om de goede woorden uit de lijst te halen en vervolgens map om elk woord om te zetten naar een tuple.		
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 tuple (filter ok words)
where ok :: String -> Bool
ok x = ...
tuple :: String -> (String, Int)
tuple 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 tuple 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":

```
["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" groter is dan MAX_INT [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	Nakijker: rkoot	Cijfer: 8.8
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is voldoende becommentarieerd. [OPGAVE 1] Dit had efficiënter geïmplementeerd 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] Werkt, maar niet erg efficiënt. [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 gemaakt van standaard- en hulpfuncties. Code is matig becommentarieerd. [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

<p>Omschrijving:</p> <p>-De gray-codering is door de oneindige aanpak wat traag, een directe formule was beter geweest.</p> <p>+Je hebt goed gebruik gemaakt van high order functions. Je had deze ook kunnen toepassen bij de eerste opgaven!</p> <p>+Het commentaar is goed en effectief.</p> <p>+De code-stijl is netjes.</p> <p>+De look-and-say reeksen zijn oneindig opgebouwd.</p> <p>+De Keith getallen zijn oneindig opgebouwd.</p> <p>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 vise-versa 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!</p>		
rgroot	Nakijker: bschuur	Cijfer: 9.0
<p>Omschrijving:</p> <p>+De gray-codering is zonder oneindige lijsten opgelost.</p> <p>+De look-and-say reeksen zijn oneindig opgebouwd.</p> <p>+De Keith getallen zijn oneindig opgebouwd.</p> <p>+De code-stijl is netjes.</p> <p>+Het commentaar is goed en effectief.</p> <p>-Je had meer high order functions kunnen gebruiken (folds i.p.v. recursie, lijstcomprehensie).</p> <p>Opgave 6:</p> <p>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.</p>		
fsteeg-hkbarnev	Nakijker: rkoot	Cijfer: 9.0
<p>Omschrijving:</p> <p>[ALGEMEEN]</p> <p>Code is duidelijk leesbaar.</p> <p>Handig gebruik gemaakt van standaard- en hulpfuncties.</p> <p>Code is goed becommentarieerd.</p> <p>[OPGAVE 1]</p> <p>Prima</p> <p>[OPGAVE 2]</p> <p>Prima.</p> <p>[OPGAVE 3]</p> <p>Prima.</p> <p>[OPGAVE 4]</p> <p>Prima.</p> <p>[OPGAVE 5]</p> <p>Prima.</p> <p>[OPGAVE 6]</p> <p>Prima.</p> <p>[OPGAVE 7]</p> <p>Werkt niet altijd correct, bijvoorbeeld voor numbers 18 ["foqnae","i","hts","xgvb","tkcmd","erhmt","l","qdoooq"]</p> <p>[OPGAVE 8]</p> <p>Werkt niet altijd correct, bijvoorbeeld voor het getal 216 in radix 8.</p> <p>[OPGAVE 9]</p> <p>Prima.</p> <p>[OPGAVE 10]</p> <p>Prima.</p>		
mlmbroer-jmwbrete	Nakijker: svisser	Cijfer: 9.1
<p>Omschrijving:</p> <p>[ALGEMEEN]</p> <p>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.</p> <p>[OPGAVE 1]</p> <p>OK.</p>		

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
fromDec = 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.

mkroese	Nakijker: svisser	Cijfer: 9.2
---------	-------------------	-------------

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.

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 ;-)

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.

pqgroot	Nakijker: svisser	Cijfer: 9.3
---------	-------------------	-------------

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.

[OPGAVE 8]
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 $n \leq \text{base}$.

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: $\text{base} - 1$).

Ook ontbrak de hoofdfunctie grayCode hoewel deze eenvoudig geschreven

kan worden als combinatie van toGrayCode en fromGrayCode.

[OPGAVE 9]
OK.

[OPGAVE 10]
OK.

Probeer gemeenschappelijke expressies in een where te zetten, dit 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 beperkt door `Int` te gebruiken.
+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.
+Het commentaar is goed en effectief.

Erg goed gewerkt je hebt een goede en correcte code-stijl. Je code is al netjes compact, met hogere orde functies kan je dit nog verder verbeteren.

wlelsing

Nakijker: bschuur

Cijfer: 9.5

Omschrijving:

-De gray-codering is door de recursieve aanpak wat traag en zorgt voor overflows.
+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.

Goed werk! Veel higher orde runcties gebruikt, je zou ze nog meer kunnen gebruiken voor de eerste opgave om die nog net iets strakker te maken.

ergallo

Nakijker: rkoot

Cijfer: 9.5

Omschrijving:

[ALGEMEEN]
Uitstekend!
Code is duidelijk leesbaar.
Handig gebruik gemaakt 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 inefficiënt.

[OPGAVE 9]
Prima.

[OPGAVE 10]
Prima.

mgrimme

Nakijker: svisser

Cijfer: 9.5

Omschrijving:

[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 tuple).

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

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
Omschrijving: [ALGEMEEN] Code is duidelijk leesbaar. Handig gebruik gemaakt van standaard- en hulpfuncties. Code is goed becommentarieerd.		
[OPGAVE 1] Prima.		
[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.		