INFSEN01-1 Opdracht 1

# Opdracht

Doel van deze opgave is het verwerven van vaardigheid in het werken met lijsten. Maak in Haskell onderstaande functies en test de oplossing.

Onder een decimaal getal in een String verstaan we een zo lang mogelijke aaneengesloten rij cijfers. De String "abc123 rtl45" bevat dus twee decimale getallen: "123" en "45".

1. containsDecimal :: String -> Bool  
   containsDecimal xs is true als xs een decimaal getal bevat, anders false.
2. getDecimals :: String -> [Int]  
   getDecimals xs geeft een lijst terug met de gevonden decimalen
3. decimalCount :: String -> Int  
   decimalCount xs is het aantal decimale getallen in xs
4. longestDecimal :: String -> Int  
   longestDecimal xs is de lengte van het langste decimale getal in xs
5. maxDecimal :: String -> Integer  
   maxDecimal xs is het grootste decimale getal in s
6. Een functie die, gegeven twee lijsten een lijst teruggeeft bestaande uit de elementen die in beide lijsten voorkomen. Let op: elk element mag slechts éénmaal worden opgenomen. Voorbeeld:   
     
   intersection “programming” “software” => “roa”
7. Een functie die, gegeven twee lijsten een lijst teruggeeft bestaande uit de elementen die in één van beide lijsten voorkomen, maar niet in beide. Let op: elk element mag slechts éénmaal worden opgenomen. Voorbeeld:  
     
   disjunction “programming” “software” =>“pgminsftwe”
8. Een functie die, gegeven twee lijsten teruggeeft of de eerste lijst een permutatie is van de tweede. Voorbeeld:   
     
   isPermutation “programming” “software” => False  
   isPermutation [1,4,6] [4,1,6] => True
9. Een functie die, gegeven twee Strings een String teruggeeft van de tekens die in de eerste rij precies even vaak voorkomen als in de tweede rij. Let op: elk teken in het resultaat mag maar éénmaal worden opgenomen. Voorbeeld:  
     
   equalCount “programming” “software” => “oa”
10. Een functie elken die, gegeven een lijst, een lijst van lijsten maakt. De eerste lijst in de output is gelijk aan de input, de tweede bevat elk 2e element. De nde lijst bevat elk nde element uit de input.  
      
    elken "ABCD" == ["ABCD", "BD", "C", "D"]  
    elken "hello!" == ["hallo!", "al!", "l!", "l", "o", "!"]  
    elken [1] == [[1]]  
    elken [True,False] == [[True,False], [False]]  
    elken [] == []
11. Een functie die, gegeven een lijst, een lijst met een lokaal maxima als output heeft. Een lokaal maximum is een element dat groter is dan het voorgaande en volgende element. Bijvoorbeeld in de lijst [2,3,4,1,5] is 4 het enige lokaal maximum. Immers 4 is groter dan 3 en 1 en na 5 volgt geen element.  
      
    locMax [2,9,5,6,1] == [9,6]  
    locMax [2,3,4,1,5] == [4]  
    locMax [1,2,3,4,5] == []

# INLEVEREN

Uiterlijk 19 maart 2015 om 17.00 digitaal via N@tschool. Als dit niet lukt kan via email met in het onderwerp de volgende regel:

INFSEN01-1 - ASSIGNMENT-1, <studentnr> <naam> [<studentnr> <naam>]

Het ingeleverde materiaal bevat een Haskell script met de functies en een document met daarin de testen. In het script staan (uiteraard) de namen van de studenten die aan deze opdracht hebben gewerkt. Opdrachten die niet voor eerder genoemd tijdstip zijn opgeleverd worden niet meegenomen in de beoordeling. Vanzelfsprekend, zoals in alle code, is het script zodanig gestructureerd en voorzien van commentaar dat iemand anders snel kan achterhalen wat er gebeurd. Elke functie is voorzien van een signatuur.