INFSEN01-1 - oefenopdracht week 4

## Opdracht 1

Definieer onderstaande functies.

1. containsDigit :: String -> Bool  
   containsDigit xs geeft true als String een cijfer bevat.
2. digitCount :: String -> Int  
   digitCount xs is het aantal cijfers dat in String voorkomt.
3. digitIndex :: String -> Int  
   digitIndex xs is de eerste index van String waar een cijfer staat en 1 als er geen cijfer in s staat.
4. evenCount :: [Int] -> Int  
   evenCount xs is het aantal even getallen in xs
5. containsEven :: [Int] -> Bool  
   containsEven xs is true, als xs een even getal bevat, anders false
6. allEven :: [Int] -> Bool  
   allEven xs is true, als alle elementen van xs even zijn, anders false
7. evenSum :: [Int] -> Int  
   evenSum xs is de som van de even getallen uit xs
8. containsSquare :: [Int] -> Bool  
   containsSquare xs is true, als xs een kwadraat bevat
9. isOrdinal :: [a] -> Bool  
   isOrdinal xs is true als elk element (behalve het laatste) kleiner is dan zijn opvolger
10. primeCount :: [Int] -> Int  
    primeCount xs is het aantal priemgetallen in xs

## Opdracht 2

Definieer onderstaande functies recursief zonder naar de standard prelude te kijken:

1. Bepaal of alle logische waarden in een lijst true zijn:  
     
   en :: [Bool] -> Bool
2. Concateneer een lijst van lijsten:  
     
   concateneer :: [[a]] -> [a]
3. Maak een lijst met gelijke elementen:  
     
   repliceer :: Int -> a -> [a]
4. Selecteer element uit een lijst:  
     
   (!!) :: [a] -> Int -> a
5. Bepaal of een waarde is in de lijst  
     
   element :: Eq a => a -> [a] -> Bool

## Opdracht 2

Definieer een recursieve functie

voegsamen :: [Int] -> [Int] -> [Int]

Die twee (2) gesorteerde lijsten van integers samenvoegt tot één (1) gesorteerde lijst van integers.

voegsamen [2,6,9] [1,5,8,12]  
[1,2,5,6,8,9,12]

## Opdracht 3

Definieer een recursieve functie:

mengsort :: [Int] -> [Int]

waarvoor geldt:

* lijsten met lengte <= 1 zijn gesorteerd;
* andere lijsten kunnen gesorteerd worden door beide helften van de lijst te sorteren en deze samen te voegen;

## Opdracht 4

Schrijf de comprehension [f x | x  xs, p x] met behulp van map en filter functies.

## Opdracht 5

Schrijf de map f en filter p met behulp van foldr