

Funktionale Programmierung

4. Übungsblatt (für das Tutorium)

Prof. Dr. Margarita Esponda

Thema: Listengeneratoren, endrekursive Funktionen und Funktionen höherer Ordnung.

1. Aufgabe

Definieren Sie eine Funktion **maxNeighborsDistance**, die den größten Abstand zwischen zwei benachbarten Zahlen einer Liste findet.

Anwendungsbeispiel:

`maxNeighborsDistance [3, 6, 2, 0, 1, 9, 8, 5] => 8`

2. Aufgabe

Schreiben Sie eine Funktion **allPairMults**, die unter sinnvoller Verwendung von Listengeneratoren die Liste mit den Produkten aller Zweier-Zahlenkombinationen einer Eingabeliste berechnet. Die Elemente der Liste können mit sich selber kombiniert werden.

Anwendungsbeispiel:

`allPairMults [2, 3, 1] => [4, 6, 2, 6, 9, 3, 2, 3, 1]`

3. Aufgabe

Betrachten Sie folgende Funktionsdefinitionen:

```
okt2bin :: [Char] -> [Char]
okt2bin [] = []
okt2bin (x:xs) = okt2Bits x ++ (okt2bin xs)
  where
    okt2Bits z | '0' <= z && z < '8' = oktTable !! (digitToInt z)
              | otherwise         = error "not octal digits"

oktTable = ["000", "001", "010", "011", "100", "101", "110", "111"]

ggt_of :: [Integer] -> Integer
ggt_of zs | (length zs) < 2 = error "The function is only defined for 2 or more numbers"
          | otherwise       = aux zs
  where
    aux [0] = error "ggt_of is not defined for a list of zeros"
    aux [x] = x
    aux (x:y:xs) | x /= 0 = aux ((ggt x y):xs)
                  | otherwise = aux (y:xs)

ggt :: Integer -> Integer -> Integer
ggt a b | a == 0 && b == 0 = error "not defined for 0 0"
        | b == 0         = a
        | otherwise       = ggt b (mod a b)
```

Definieren Sie unter sinnvoller Verwendung der **map**, **filter**, **foldr** oder **foldl** Funktionen folgende Funktionen.

```
okt2bin :: [Char] -> [Char]
ggt_of :: [Integer] -> Integer
```