

## Übungsblatt 13

Abgabe bis Freitag, 08.02.2013, 12:00 Uhr

**Hinweis:** Lösungen immer per Email an den zuständigen Tutor schicken. Die Emailadressen sind auf der Homepage zur Vorlesung gelistet.

Auf der folgenden Homepage können Sie an der Lehrevaluation für die Vorlesung “Einführung in die Informatik” teilnehmen. [http://www.informatik.uni-freiburg.de/~welte/lehrevaluation/ws201213/webseite\\_links.html](http://www.informatik.uni-freiburg.de/~welte/lehrevaluation/ws201213/webseite_links.html)

### Aufgabe 13.1

Implementieren Sie eine Haskell-Funktion `p`, die die Signatur

```
p :: Int -> [[Int]]
```

besitzt. Diese Funktion soll für eine Zahl  $x$  alle Listen  $[a, b, c]$  zurückgeben für die alle folgenden Bedingungen gelten:

- $0 < a < b < c$ ,
- $a^2 + b^2 = c^2$  **und**
- $a + b + c = x$ .

### Aufgabe 13.2

1. Implementieren Sie eine Haskell-Funktion `myfunction`, die die Signatur

```
myfunction :: Int -> Int
```

besitzt. Diese Funktion soll das Produkt aller Teiler der Zahl, die kleiner als die Zahl selbst sind, zurückgeben.

Beispiel:

```
myfunction 15 ergibt  $1 \cdot 3 \cdot 5 = 15$ 
```

2. Berechnen Sie alle Zahlen  $< 1000$  für die gilt, dass das Produkt aller Teiler die Zahl selbst ergibt.

### Zusatzaufgabe: Aufgabe 13.3

1. myfunction Implementieren Sie eine Haskell-Funktion `myLCM`, die die Signatur `myLCM :: Int -> Int -> Int` besitzt. Diese Funktion soll das kleinste gemeinsame Vielfache zurückgeben.  
Beispiel:  
`myLCM 4 10` ergibt 20
2. Implementieren Sie eine Haskell-Funktion `myLCM1`, die die Signatur `myLCM1 :: Int -> Int` besitzt. Diese Funktion soll das kleinste gemeinsame Vielfache von allen Zahlen von 1 bis zum Eingabewert zurückgeben.

### Zusatzaufgabe: Aufgabe 13.4

1. Implementieren Sie eine Haskell-Funktion `insert` mit der Signatur `insert :: (a->a->Bool) -> a -> [a] -> [a]`, die ein Element in eine sortierte Liste einfügt. Der Eingabeparameter `(a->a->Bool)` definiert dabei die Ordnungsrelation, bezüglich der die Liste sortiert sein soll.  
Beispiel: `insert (<=) 4 [1,3,7,11]` ergibt `[1,3,4,7,11]`
2. Implementieren Sie eine Haskell-Funktion `isort` mit der Signatur `isort :: (a->a->Bool) -> [a] -> [a]`, die eine Liste bezüglich der angegebenen Ordnungsrelation sortiert. Verwenden Sie dafür `insert` und die Haskell-Funktion `foldr`.