

(FP-Aufgaben 07)

Aufgabe 1: Was ist das Ergebnis von (f 5)?

```
f = \ x -> (4 * x + 1) `div` 2
```

Aufgabe 2: Welchen Typ haben die folgenden Funktionen f1 und f2 ?

```
f1 = (++[1, 2, 3])
```

```
f2 x y = x : f1 y
```

Aufgabe 3: Verwenden Sie die Akkumulator-Technik und schreiben Sie eine Funktion, die eine Liste von Integer-Zahlen umkehrt (z.B. wäre die Liste [1,6,3,8], dann sollte das Ergebnis [8,3,6,1] sein).

Aufgabe 4: Verwenden Sie die foldr-Funktion und definieren Sie die Funktion f, die die folgende Summenformel ausrechnet. Dabei sind i und n ganze positive Zahlen.

$$f(n) = \sum_{i=1}^n i^2$$

Hinweis: Vervollständigen Sie den folgenden Code:

```
f :: Int -> Int  
f n = foldr _____ [1..n]
```

Aufgabe 5: Die Daten der Studierenden sind in eine Liste eingetragen. Jeder Eintrag der Liste besteht aus einer „Matrikelnummer“ und einem „Notenkonto“. Das Notenkonto ist selbst eine Liste. Jeder Eintrag vom Notenkonto besteht aus einem „Fach“ und einer „Note“:

```
liste :: [ ( Integer , [ ( String , Double ) ] ) ]
```

Definieren Sie eine Funktion, die eine solche Liste und die Matrikelnummer eines Studenten als Parameter nimmt und ausgibt wie viele Prüfungen der Student bestanden hat (Benutzen Sie dabei keine vordefinierten Funktionen. Sie dürfen aber eigene Hilfsfunktionen definieren).

Hinweis:

- Zuerst suchen Sie nach dem Notenkonto des Studenten:

```
gibNotenkonto :: [(Integer, [(String, Double)])] -> Integer -> [(String, Double)]
```
- Dann rechnen Sie die Anzahl der Prüfungen, die eine Note zwischen 1 und 4 haben:

```
bestanden :: [(String, Double)] -> Int
```