

**Aufgaben zu Listen in Haskell – rekursiv -****1. Aufgabe - grundlegende Listenfunktionen – selbst nachgebaut****a) Funktion `laenge`**

Länge einer Liste zurückgeben (`length` in Haskell)

**b) Funktion `letztes`**

Letztes Element einer Liste zurückgeben (`last` in Haskell)

**c) Funktion `anfang`**

Anfang einer Liste zurückgeben (`init` in Haskell). Liefert die Liste ohne das letzte Element.

**d) Funktion `gibNtes`**

Gibt das n-te Element einer Liste zurück (`!!` in Haskell). Die Zählung beginnt üblicherweise bei Null.

**e) Funktion `verbinde`**

Hierbei wird die vordere Liste nach und nach in ihre Einzelteile zerlegt und dann an die zweite Liste „angeconst“ (`++` in Haskell). Beide Listen sind dann zu einer verbunden (concateniert).

**2. Aufgabe – weitere Funktionen auf Listen**

a) `quadlist liste` - Quadrate der Listenelemente, die natürlich nur Zahlen sein können

b) `istEnthalten x liste` - testet, ob x in der Liste ist

c) `delErstes x liste` - löscht das erste Auftreten von x in der Liste

d) `delElem x liste` - löscht alle x aus der Liste

e) `alleGleich liste` - testet, ob alle Elemente der Liste gleich sind

f) `streichGleiche liste` - löscht alle Mehrfachvorkommen von Elementen aus der Liste, also

```
>streicheGleiche "Hallohoho"  
>"Halo"
```

Hierdurch wird aus einer Liste eine Menge. In einer Menge kommt jedes Element nur einmal vor!

Die folgende Aufgabe beziehen sich auf solche Mengen.

**3. Aufgabe – weitere Funktionen auf Listen (Mengen)**

a) `vereinigung menge1 menge2` - Vereinigungsmenge beider Mengen

b) `durchschnitt menge1 menge2` - Durchschnittsmenge beider Mengen

**4. Aufgabe – weitere Funktionen auf beliebige Listen**

a) `insert x liste` - schreibt x zwischen alle Listenelemente

b) `replaceAllElem x y liste` - ersetzt jedes Auftreten von x durch y

c) `verdoppleElem liste` - verdoppelt alle Elemente, also

```
>verdoppleElem [1,2,3]  
>[1,1,2,2,3,3]
```

d) `verdoppleListe liste` - verdoppelt die Liste,

```
>verdoppleListe [1,2,3]  
>[1,2,3,1,2,3]
```