

# 1 Listen

## 1.0.1 Aufgabe

Schreibe eine Funktion `grow`. Dieser wird eine Liste mit Integer übergeben. Sie gibt das Produkt der Integer zurück.

```
1 grow(listOf(1 , 2, 2))
```

4

```
1 grow(listOf(13, 2))
```

26

<https://www.codewars.com/kata/57f780909f7e8e3183000078/train/kotlin>

## 1.0.2 Aufgabe

Schreibe eine Funktion `squareSum`. Dieser wird eine Liste von Integer übergeben. Sie gibt die Summe der Quadrate der Zahlen in dieser Liste zurück. Z.B gilt

$$\text{square\_sum}(\text{listOf}(1,2)) = 1^2 + 2^2 = 5$$

```
1 squareSum(listOf(1, 2))
```

5

```
1 squareSum(listOf(0, 3, 4, 5))
```

50

<https://www.codewars.com/kata/515e271a311df0350d00000f/train/kotlin>

## 1.0.3 Aufgabe

Schreibe eine Funktion `positiveSum`. Dieser wird eine Liste von Integer übergeben. Sie gibt die Summe der positiven Zahlen in dieser Liste zurück.

```
1 positiveSum(listOf(1, -2, 3, -4, 5))
```

9

```
1 positiveSum(listOf(1, -2, -3, 4, -5))
```

5

<https://www.codewars.com/kata/5715eaedb436cf5606000381/train/kotlin>

#### 1.0.4 Aufgabe

Schreibe eine Funktion `findSmallestInt`. Dieser wird eine nichtleere Liste von Integern übergeben. Sie gibt das kleinste Integer in der Liste zurück.

```
1 findSmallestInt(listOf(1, 2, 3))
```

1

```
1 findSmallestInt(listOf(1, 0, -5))
```

-5

<https://www.codewars.com/kata/55a2d7ebe362935a210000b2/train/kotlin>

#### 1.0.5 Aufgabe

Schreibe eine Funktion `smallEnough`. Dieser wird eine Liste von Integern und ein weiteres Integer übergeben. Falls alle Integer in der Liste kleiner oder gleich groß wie diese Zahl sind wird `true` zurückgegeben. Ansonsten wird `false` zurückgegeben.

```
1 smallEnough(listOf(1, 2, 3), 3)
```

true

```
1 smallEnough(listOf(1, 2, 3), 2)
```

false

<https://www.codewars.com/kata/57cc981a58da9e302a000214/train/kotlin>

#### 1.0.6 Aufgabe

Schreibe eine Funktion `maps`. Dieser wird eine Liste von Integern übergeben. Sie gibt eine Liste zurück in der jeder Eintrag der übergebenen Liste verdoppelt wurde.

```
1 maps(listOf(1, 2, 3))
```

```
[2, 4, 6]
```

```
1 maps(listOf(13, 9))
```

```
[26, 18]
```

<https://www.codewars.com/kata/57f781872e3d8ca2a000007e/train/kotlin>

### 1.0.7 Aufgabe

Schreibe eine Funktion `invert`. Dieser wird eine Liste von Integern übergeben. Sie gibt eine Liste zurück in der das Vorzeichen jedes Eintrags der übergebenen Liste umgedreht wurde.

```
1 invert(listOf(1, 2, 2))
```

```
[-1, -2, -2]
```

```
1 invert(listOf(1, 4, 0))
```

```
[-1, -4, 0]
```

<https://www.codewars.com/kata/5899dc03bc95b1bf1b0000ad/train/kotlin>

### 1.0.8 Aufgabe

Schreibe eine Funktion `points`. Dieser wird eine Liste von Strings übergeben. Jeder String steht für Das Ergebnis eines Fußballspiel z.B. `"3:2"`. Der erste Buchstabe ist die Punktzahl von unserem Team. Dann folgt immer ein Doppelpunkt und die Punktezahl des anderen Teams. Die Funktion gibt die Punktezahl unseres Teams nach allen Spielen in der Liste zurück. Du kannst annehmen, dass kein Team mehr als 9 Tore in einem Spiel geschossen hat. Bei eines Sieg bekommt ein Team 3 Punkte, bei einem Unentschieden einen Punkt und bei einer Niederlage gar keine Punkte.

```
1 points(listOf("1:0", "2:0"))
```

```
6
```

```
1 points(listOf("1:0", "2:3", "2:2"))
```

```
4
```

<https://www.codewars.com/kata/5bb904724c47249b10000131/train/kotlin>

### 1.0.9 Aufgabe

Schreibe eine Funktion `twoOldestAges`. Dieser wird eine Liste mit (mindestens zwei) Integern übergeben. Sie soll eine Liste zurückgeben, die die beiden höchsten Zahlen in der Liste enthält.

```
1 twoOldestAges(listOf(1, 5, 87, 45, 8, 8))
```

```
[45, 87]
```

```
1 twoOldestAges(listOf(10, 1))
```

```
[1, 10]
```

<https://www.codewars.com/kata/511f11d355fe575d2c000001/train/kotlin>

### 1.0.10 Aufgabe

Schreibe eine Funktion `tribonacci`. Dieser wird eine Liste und eine weitere Zahl  $n$  übergeben. Die Funktion soll die ersten  $n$ -te Zahlen der Tribonacci-Folge als Liste zurückgeben.

Die übergebene Liste enthält die ersten 3 Zahlen dieser Folge. Jede weitere Zahl ist die Summe ihrer 3 Vorgänger.

```
1 tribonacci(listOf(1, 2, 3), 5)
```

```
[1, 2, 3, 6, 11]
```

```
1 tribonacci(listOf(1, 0, 3), 2)
```

```
[1, 0]
```

<https://www.codewars.com/kata/556deca17c58da83c00002db/train/kotlin>

### 1.0.11 Aufgabe

Schreibe eine Funktion `findOutlier`. Dieser wird eine Liste von mindestens 3 Integern übergeben. Die Liste enthält

1. entweder nur gerade Zahlen und eine ungerade Zahl. In diesem Fall soll die ungerade Zahl zurückgegeben werden
2. oder nur ungerade Zahlen und eine gerade Zahl. Dann soll die gerade Zahl zurückgegeben werden

```
1 findOutlier(listOf(2, 4, 6, 8, 10, 3))
```

3

```
1 findOutlier(listOf(160, 3, 1719, 19, 11, 13, 21))
```

160

<https://www.codewars.com/kata/5526fc09a1bbd946250002dc/train/kotlin>

## 2 Veränderbare Listen

### 2.0.1 Aufgabe

Schreibe eine Funktion `reverseSeq`. Dieser wird ein Integer übergeben. Sie gibt alle Zahlen von diesem Integer bis zur 1 in absteigender Reihenfolge zurück.

```
1 reverseSeq(5)
```

[5, 4, 3, 2, 1]

```
1 reverseSeq(1)
```

[1]

<https://www.codewars.com/kata/5a00e05cc374cb34d100000d/train/kotlin>