

# 1 While-Schleifen

## 1.1 Aufgabe

Implementiere eine Funktion `powerOfTwo`, die prüft, ob eine nicht negative ganze Zahl eine Zweierpotenz ist.

```
1 powerOfTwo(0)
```

false

```
1 powerOfTwo(1)
```

true

```
1 powerOfTwo(2)
```

true

```
1 powerOfTwo(5)
```

false

```
1 powerOfTwo(16)
```

true

## 1.2 Aufgabe

Implementiere eine Funktion `halvingSum`, die für eine positive Zahl  $n$  die Summe

$$n + n/2 + n/4 + \dots + 1$$

berechnet. Z.B. gilt `halving_sum(25) = 25 + 12 + 6 + 3 + 1 = 47`.

```
1 halvingSum(25)
```

47

```
1 halvingSum(1)
```

1

### 1.3 Aufgabe

Implementiere eine Funktion `nbYear`, die berechnet wann eine Population mit gegebenem Anfangsbestand, jährlichem prozentualen Wachstum und jährlicher Zuwanderung eine bestimmte Grenze überschreitet.

Z.B. könnte man eine Kleinstadt mit 1000 Einwohnern, einem jährlichen Wachstum von 2 Prozent und 50 Personen, die jedes Jahr in die Stadt ziehen betrachten und sich fragen wann 1200 Personen in der Stadt wohnen.

$$1. \text{ Jahr } 1000 + \lfloor 1000 * 0.02 \rfloor + 50 = 1070 \text{ Einwohner}$$

$$2. \text{ Jahr } 1070 + \lfloor 1070 * 0.02 \rfloor + 50 = 1141 \text{ Einwohner}$$

$$3. \text{ Jahr } 1141 + \lfloor 1141 * 0.02 \rfloor + 50 = 1213 \text{ Einwohner}$$

Die Grenze ist also nach 3 Jahren erreicht.

Die Klammern um den mittleren Term stehen für eine Abrundung. Es können ja keine halben Personen dazukommen.

```
1 nbYear(1000, 2.0, 50, 1200)
```

```
3
```

<https://www.codewars.com/kata/563b662a59afc2b5120000c6/train/kotlin>

### 1.4 Aufgabe

Implementiere eine Funktion `isSquare`, die prüft, ob eine natürliche Zahl eine Quadratzahl ist.

```
1 isSquare(0)
```

```
true
```

```
1 isSquare(1)
```

```
true
```

```
1 isSquare(2)
```

```
false
```

## 1.5 Aufgabe

Collatz-Folgen werden nach dem folgenden Prinzip gebildet.

1. Wähle eine beliebige natürliche Zahl  $n$
2.
  - Wenn  $n$  gerade ist, wähle als nächstes Folgenglied  $\frac{n}{2}$
  - Wenn  $n$  ungerade ist, wähle als nächstes Folgenglied  $3 * n + 1$
3. Wiederhole Schritt 2 immer wieder bis die Zahl 1 erreicht ist.

Formaler ausgedrückt, wird die folgende Funktion auf ein Folgenglied angewendet um das nächste Folgenglied zu erhalten.

$$f(n) = \begin{cases} n/2 & \text{falls } n \text{ gerade ist} \\ 3n + 1 & \text{falls } n \text{ ungerade ist} \end{cases}$$

Z.B. erhält man für die Startzahl 5 die folgende Collatz-Folge: 5, 16, 8, 4, 2, 1

Implementiere eine Funktion `hotpo`, die berechnet wie viele der oben erklärten Berechnungen durchgeführt werden müssen, um von einer eingegeben Startzahl auf die Zahl 1 zu kommen

```
1 hotpo(1)
```

0

```
1 hotpo(5)
```

5

## 1.6 Aufgabe

Implementiere eine Funktion `collatz`. Diese gibt einen String zurück, der alle Zwischenergebnisse der Collatz-Folge anzeigt, bis 1 erreicht wurde.

```
1 collatz(1)
```

1->1

```
1 collatz(5)
```

5->1