

1 Rekursion

1.1 Rekursionen bestimmen

1. Gegeben sind die Anfangsglieder einer Folge von Zahlen:

5, 10, 20, 40, 80, 160, 320, 640, 1280, 2560, ...

Finden Sie die Rekursionsvorschrift, die eine Folge mit diesen Anfangsgliedern beschreibt. Unterstellen Sie, dass sich die Folge so weiter entwickelt, wie es diese Anfangsglieder vermuten lassen.

2. Die Summe der ersten n ungeraden Zahlen bildet die Quadratzahl von n :

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

- (a) Testen Sie die Formel an einer Folge von Zahlen.
- (b) Finden Sie eine Rekursionsvorschrift, die diese iterative Formel beschreibt.
- (c) Beweisen Sie durch vollständige Induktion die Korrektheit Ihrer rekursiven Vorschrift unter Zuhilfenahme der Gleichung (1).

1.2 Gerade oder Ungerade?

Schreiben Sie zwei rekursive Funktionen `even(n)` und `odd(n)`, die jeweils zurückgeben, ob eine übergebene natürliche Zahl n gerade oder ungerade ist. Schreiben Sie die beiden Funktionen in Python.

Damit die Aufgabe nicht allzu trivial wird, werden Sie eingeschränkt, auf den Modulo-Operator zu verzichten. Weiterhin dürfen Sie lediglich benutzen, dass 0 eine gerade Zahl ist, dass auf eine gerade Zahl eine ungerade Zahl folgt und dass auf eine ungerade Zahl wieder eine gerade Zahl folgt.

Beispiel:

```
1 even(3) == False
2 even(4) == True
3 odd(2)  == False
4 odd(5)  == True
```

Hinweis: Sie können eine rekursive Funktion in Abhängigkeit einer anderen Funktion definieren.

1.3 Die Fibonacci Zahlenfolge

Der Wert der Fibonacci Zahlenfolge $fibo(n)$ an der Stelle n ergibt sich aus der Summe der beiden vorhergehenden Werte der Folge. Die ersten Zahlen $fibo(0)$, $fibo(1)$, ..., $fibo(7)$ dieser Folge sind:

1, 1, 2, 3, 5, 8, 13, 21

Schreiben Sie eine Funktion in Python, die die n -te Fibonacci Zahl berechnet:

a) rekursiv

b) iterativ