

**Aufgabe 1** Funktionen in Python

Definieren Sie eine Funktion `revDigits`, welche die Ziffernfolge einer beliebigen natürlichen Zahl umkehrt. Zum Beispiel: `revDigits (1345001) = 1005431` und `revDigits (8390100) = 10938`.

**Aufgabe 2** Rekursionsdrill

- (a) Schreiben Sie eine rekursive Funktion, welche die Länge einer gegebenen Liste berechnet.
- (b) Schreiben Sie eine rekursive Funktion, die das Produkt einer Liste von Zahlen berechnet.
- (c) Schreiben Sie eine rekursive Funktion, die das Minimum einer Liste von Zahlen berechnet.
- (d) Schreiben Sie eine rekursive Funktion, welche die Elemente einer Liste umkehrt.
- (e) Schreiben Sie eine rekursive Funktion, welche eine natürliche Zahl  $n$  erhält, und eine Zeichenkette liefert, welche  $n$  in Binärdarstellung repräsentiert.
- (f) Schreiben Sie eine rekursive Funktion, welche eine Binärzahl, die als String gegeben ist, in eine Zahl umwandelt.
- (g) Schreiben Sie eine rekursive Funktion, welche eine Liste aller Teilmengen (als Listen) zu den Elementen einer gegebenen Liste erzeugt.
- (h) Ein Polynom vom Grad  $n$  ist eine Funktion der Form  $f(x) = \sum_{i=0}^n a_i x^i$ . Wir können  $f$  schreiben als  $f(x) = a_0 + x \cdot g(x)$ , wobei  $g$  ein Polynom vom Grad  $n-1$  ist. Benutzen Sie diese Beobachtung, um eine rekursive Funktion zu schreiben, welche ein Polynom  $f$  an einer gegebenen Stelle  $x_0$  auswertet (die Koeffizienten werden als eine Liste übergeben).
- (i) Schreiben Sie eine rekursive Funktion, welche die geometrische Summe

$$\sum_{i=0}^n a^i$$

für gegebene  $a$  und  $n$  berechnet.

- (j) Für  $x > 0$  und eine gerade natürliche Zahl  $n$  gilt:

$$x^n = \left(x^{\lfloor n/2 \rfloor}\right)^2.$$

Für  $x > 0$  und eine ungerade natürliche Zahl  $n$  gilt:

$$x^n = x \cdot (x^{\lfloor n/2 \rfloor})^2.$$

Benutzen Sie diese Beobachtung, um eine rekursive Funktion zu schreiben, die  $x^n$  zu gegebenen  $x$  und  $n$  berechnet.

### Aufgabe 3 Parameterübergabe

Was ist der Unterschied zwischen den formalen und den tatsächlichen Parametern einer Funktion? Geben Sie ein Beispiel für eine Funktion, die sich unterschiedlich verhält, wenn die Parameter per *call-by-value* und per *call-by-reference* übergeben werden.

### Aufgabe 4 Gültigkeitsbereiche und Closures

Betrachten Sie den folgenden Python-Code:

```
frage = "Wie alt sind Sie? "

def getGreeter(name):
    gruss = "Hallo "

    def greet():
        print(gruss + name + "!")
        alter = input(frage)
        return int(alter)

    for i in range(1,4):
        print(i)

    return greet

g1 = getGreeter("Katharina")
g2 = getGreeter("Max")
a1 = g1()
a2 = g2()
print("Das Gesamtalter ist " + str(a1 + a2) + ".")
```

- Markieren Sie für jede Variable im obigen Python-Code den jeweiligen Gültigkeitsbereich in Programm.
- Was genau macht die Funktion `getGreeter`? Was genau macht die Funktion `greet`? Informieren Sie sich über *Closures* in Python. Was haben Closures mit unserem Python-Beispiel zu tun?