

# Blatt 2

Vorkurs Bauingenieurwesen - Informatik - 09.03.24

## Aufgabe 1

Konvertiere  $011111101000_{bin}$  ins Dezimalsystem von Hand.

## Aufgabe 2

Konvertiere  $1365_{dec}$  ins Binärsystem von Hand.

## Aufgabe 3

Berechne  $01010100110_{bin} + 01010010011_{bin}$  im Binärsystem von Hand.

## Aufgabe 4

Konvertiere die Ergebnisse der vorherigen Aufgaben ins Hexadezimalsystem.

## Aufgabe 5

Schreibe eine Funktion `betrag`, die den Betrag einer Zahl berechnet.

```
def betrag(x: float) -> float:
```

## Aufgabe 6

Du hast dich in die NSA gehackt um amerikanische Spionageprogramme zu stehlen. Ein Programm fällt dir dabei ins Auge. Anscheinend hat ein NSA Praktikant versucht eine Funktion zu schreiben, die die  $n$ -te Fibonaccizahl berechnet. Die nullte Fibonaccizahl ist 0, die erste Fibonaccizahl ist 1, jede weitere Fibonaccizahl ist die Summe der beiden vorherigen. Die Folge beginnt also mit 0, 1, 1, 2, 3, 5, 8, 13, .... Wieso funktioniert die Funktion des Praktikanten so nicht?

```
### TOP SECRET ###
def fibonacci(n: int) -> int:
    a = 0 # zeroth fibonacci number
    b = 1 # first fibonacci number
    while n > 0:
        # advance both a and b to next
        a = b
        b = a + b
        # repeat n times
        n = n - 1
    # now a is the n-th fibonacci number
    return a
```

Korrigiere die Funktion.

## Aufgabe 7

Schreibe eine Funktion `int_zu_binaer`, die die Binärziffern einer nicht-negativen ganzen Zahl als `str` zurückgibt.

```
def int_zu_binaer(n: int) -> str:
```

Zwei Zeichenketten (Variablen vom Typ `str`) können mit `+` aneinandergehängt werden. Eine Zahl `n: int` kann von python mit `str(n)` in die Zeichenkette ihrer Dezimalziffern umgewandelt werden. Die Ausgabe des folgenden Codes ist `Hall10`.

```
n = 0
s1 = "Hall"
s2 = s1 + str(n)
print(s2)
```

## Aufgabe 8

Schreibe eine Funktion `int_zu_dezimal`, die die Dezimalziffern einer nicht-negativen ganzen Zahl als `str` zurückgibt.

```
def int_zu_dezimal(n: int) -> str:
```

Verboten ist dabei die `str` Funktion auf Zahlen anzuwenden, die mehr als eine Dezimalziffer besitzen.

## Aufgabe 9

Schreibe eine Funktion die mithilfe eine while-Schleife berechnet, was die kleinste Zweierpotenz  $\varepsilon \in \text{float}$  ist, so dass  $1.0 + \varepsilon \neq 1.0$ .

```
def float_epsilon() -> float:
```

## Aufgabe 10

Die eulersche Zahl  $e$  ist ca. 2.7182818284590452353602. Exakt lässt Sie sich berechnen als

$$e = \sum_{k=0}^{\infty} \frac{1}{k!} = \frac{1}{1} + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{1 \cdot 2 \cdot 3 \cdot 4} + \dots$$

Schreibe eine Funktion `euler`, die eine Näherung der Zahl als `float` berechnet.

```
def euler() -> float:
```