

# Programmierung 1

Übungsblatt Woche 6 - 26. November - 02. Dezember 2025

## 1. Rekursive Implementierung (blatt6\_1.c)

Die folgenden Funktionen sollen in C rekursiv implementiert werden:

- (a) Summationsfunktion:  $\sum n = 1 + 2 + \dots + (n - 1) + n$   
`int sum (int n); // Summenbildung von 1 bis n`
- (b) Fakultät:  $n! = n * (n - 1) * \dots * 1$   
`int fakultaet (int n); // Fakultaetsberechnung für n`
- (c) Fibonacci-Zahlen:  $f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2}$   
`int fibonacci (int n); // n-te Fibonacci-Zahl`

## 2. Ziffern zählen (blatt6\_2.c)

Schreiben Sie eine Funktion, der eine natürliche Zahl übergeben wird und die deren Ziffern zählt. Erstellen Sie sowohl eine iterative als auch eine rekursive Variante und testen Sie beide.

*Beispiel:* `countDigits(1234) = 4`

## 3. Klassifikation von Zeichen (blatt6\_3.c, chars.c und chars.h)

Schreiben Sie eine kleine Bibliothek, die für die Klassifikation von `chars` verwendet werden kann. Definieren Sie die folgenden Funktionen:

- `int isUpper(char c) →` gibt zurück, ob es sich um einen Großbuchstaben handelt
- `int isLower(char c) →` gibt zurück, ob es sich um einen Kleinbuchstaben handelt
- `int isDigit(char c) →` gibt zurück, ob es sich um eine Zahl handelt

Legen Sie eine passende Header-Datei an und dokumentieren Sie Ihre Schnittstelle! Erstellen Sie eine zusätzliche Datei (`blatt6_3.c`) mit einer Main-Funktion, die Ihre Bibliothek entsprechend verwendet und testet.

*Hinweis:* Schauen Sie sich die Wertebereiche von Zahlen, Groß- und Kleinbuchstaben in der ASCII-Tabelle an. `chars` sind eigentlich nur die Indizes in der Tabelle, also Zahlen, und können deshalb auch mit anderen Zahlen verglichen werden.

## 4. Hochladen und Vorstellen

Laden Sie bis spätestens Dienstag, den 02. Dezember 2025, 23:59 Uhr, die Dateien `blatt6_1.c`, `blatt6_2.c`, `blatt6_3.c`, `chars.c` und `chars.h` im eLearning hoch.