

# Übungsserie 4

Abgabe: Mittwoch 24.10.2024

*Damit für alle die gleichen Regeln gelten, müssen Ihre Programme standardkonform und ohne Warnungen mit den Compileroptionen (GCC, Clang, MSVS vergleichbar): -Wall -Wextra -Werror -pedantic -std=c11 ohne Warnungen oder Fehler kompilieren. Vermeiden Sie insbesondere nicht autorisierte Bibliotheken oder Build-Environments. Reichen Sie Ihre Programme (Bitte nur die Quelltext-Dateien) als ZIP Archiv via Opal ein.*

In diesem Übungsblatt sollen Sie Ihre Kenntnisse über Schleifen und Funktionen festigen. Arbeiten Sie die Aufgaben Schritt für Schritt durch und achten Sie dabei auf eine saubere Strukturierung des Codes und korrekte Bereichsprüfungen.

## Aufgabe 1: Zeichnen des Hauses vom Nikolaus

Schreiben Sie eine Funktion, die das Haus vom Nikolaus parametrisiert mit  $n$  Sternen in der Breite und Höhe ausgibt. Zusätzlich soll das Dach des Hauses  $n$  Sterne in der Höhe haben. Der Benutzer gibt den Wert von  $n$  ein, und die Funktion soll das Haus entsprechend dieser Parameter zeichnen.

- Implementieren Sie eine Funktion, die ein Haus vom Nikolaus mit einer variablen Anzahl von Sternen in der Breite und Höhe ausgibt.
- Testen Sie die Funktion mit verschiedenen Werten für  $n$ .

## Aufgabe 2: Skalarprodukt und Norm eines Vektors

Schreiben Sie zwei Funktionen:

- Eine Funktion, die das Skalarprodukt zweier statisch allozierter Arrays gleicher Länge berechnet. Prüfen Sie vor der Berechnung, ob die Länge der Arrays korrekt ist.
- Eine Funktion, die die Norm eines Vektors berechnet und damit den Vektor (aka statisches Array) normiert.

## Aufgabe 3: Binäre Suche in einem sortierten Array

Gegeben ist ein sortiertes Array von Ganzzahlen. Schreiben Sie eine Funktion, die eine binäre Suche darauf durchführt, um zu überprüfen, ob eine bestimmte Zahl im Array enthalten ist. Die Funktion soll den Index der gesuchten Zahl zurückgeben, falls sie gefunden wird, oder  $-1$ , falls sie nicht im Array vorhanden ist.

- Implementieren Sie die binäre Suche iterativ.
- Testen Sie die Funktion mit einem sortierten Array und verschiedenen Zielwerten.

Angenommen, das folgende Array ist gegeben:

```
const unsigned int n = 8;
int arr[] = {1, 3, 5, 7, 9, 11, 13, 15};
int target = 7;
```

In diesem Fall sollte die Funktion den Index 3 zurückgeben, da die Zahl 7 im Array an dieser Position steht. Der Sinn hinter dieser Aufgabe ist es, mit maximal  $\log n$  vielen Vergleichen den Index zu finden.