

# Inflng

Programmierpraktikum

# Aufgabe 1 - Felder

Einfacher	Mittel	Schwerer
<p>Schreibe ein Programm, bei dem jedes dritte Element eines Arrays ausgegeben wird.</p> <p>Dann: Schreibe eine Funktion, die als Parameter das Array von Integern und einen Int n übergeben bekommt und jedes n-te Element des Arrays ausgibt (cout)</p> <p>Bsp.: [4,2,5,1,6,8,3,2,5], n = 4 Ausgabe: "1,2"</p>	<p>Schreibe eine Funktion, die ein Array von Integern übergeben bekommt, wobei das gesamte Array eine Zahlenfolge in aufsteigender Reihenfolge ist mit Schrittgröße 1. Die Funktion soll den einzigen fehlenden Wert in der Zahlenfolge zurückgeben.</p> <p>Bsp.: [3,5,6,7] -&gt; 4</p>	<p>Schreibe eine Funktion, die ein Array von Integern übergeben bekommt, wobei das gesamte Array eine unsortierte Zahlenfolge ist, bei der eine beliebige Anzahl an Werten fehlen kann. Die Schrittgröße ist immer noch 1. Die Funktion soll alle fehlenden Werte in der Konsole ausgeben und die Anzahl der fehlenden Werte zurückgeben.</p> <p>Bsp.: [8,10,11,7,3,15] -&gt; 8 Mit Konsolenausgabe "4,5,6,9,12,13,14"</p>

# Aufgabe 2 - Rekursion

Einfacher	Mittel	Schwerer - Backtracking
<p>Schreibe eine Funktion, die mittels Rekursion die Fakultät einer Ganzzahl berechnet. Die Funktion soll das Ergebnis als Rückgabewert zurückgeben.</p> <p><math>5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1</math> <math>0! = 1</math> <math>1! = 1</math> <math>2! = 2 \cdot 1</math></p> <p>Bsp.: <code>fakult(3)</code> -&gt; 6</p>	<p>Schreibe eine Funktion, die eine Ganzzahl übergeben bekommt und mittels Rekursion die Anzahl der Ziffern der Zahl zurückgibt.</p> <p>Bsp.: <code>ziffern(1)</code> -&gt; 1 <code>ziffern(10)</code> -&gt; 2 <code>ziffern(123)</code> -&gt; 3</p>	<p>Gegeben sei ein Array A mit Einträgen 0 und 1 und eine Startposition und eine Endposition aus x- und y-Koordinaten. Die Funktion soll true zurückgeben, wenn ein Weg von der Startposition zur Endposition existiert, bei dem man nur 1en berührt. Man darf keine Diagonalen Schritte ausführen.</p> <p>Bsp.: auf folgender Folie</p>

## Aufgabe 2 - Beispiel für schwere Aufgabe

Gegeben: int xStart = 0, int yStart = 0, int xEnd = 0, int yEnd = 3

Array, wie rechts dargestellt.

In dem Array wird der Punkt  $x = 1$ ,  $y = 2$  an Stelle  $[2*5 + 1] = [11]$  gefunden. Das liegt an der 1-dimensionalen Darstellung. 5 ist dabei die Breite, also sizeX des Feldes. sizeY ist 4.

Rückgabe: True, man kann sich vom Startpunkt oben links zum Endpunkt unten rechts bewegen unter den gegebenen Regeln.

```
int BeispielArray[] = {  
    1,0,1,1,1,  
    1,1,1,0,1,  
    0,0,0,0,1,  
    1,1,1,1,1  
};
```

# Aufgabe 3 - Pointer, Call by Reference/Value

Einfacher	Mittel	Schwerer
<p>Erstelle eine Funktion, die zwei Ganzzahlen addiert. Die Funktion soll nichts zurückgeben, sondern das Ergebnis direkt in die Variable, die als erster Parameter angegeben wurde, speichern. Also das gleiche Prinzip wie +=.</p>	<p>Schreibe eine Funktion, die alle Einträge eines übergebenen Arrays ausgibt. Du sollst keine Zählervariable nutzen, mit der du Zugriffe wie A[i] machst.</p> <p>Tipp: Pointer können um einen Eintrag nach rechts verschoben werden durch ++</p> <p>Schaffst du es in einer Zeile innerhalb der Funktion?</p>	<p>Für Fortgeschrittene oder Wiederholer aus vorherigen Jahren: Fangt mit objektorientierter Programmierung an und implementiert eine doppelt verkettete Liste mit Hilfe von Pointern und dynamischer Speicherallozierung. Implementiert auch gängige Methoden für solche Listen. (addElement etc.)</p>

# Was mache ich über die Ferien?

## Programmierübungen

- Leetcode <https://leetcode.com/problemset/all/>
- Codewars <https://www.codewars.com/>

## Brückenkurs wiederholen

- <https://www.tuhh.de/es/ce/teaching/brueckenkurs-programmieren.html>

## Eigene Projekte starten

z.B. Das Spiel Mastermind als Spiel in der Konsole implementieren