



## Aufgabenblatt 5

Ausgabe: 24.10.2022 14:00

Abgabe: 26.10.2022 22:00

**Thema:** Arrays, Call-by-Reference, Call-by-Value

### Abgabemodalitäten

1. Die Aufgaben des C-Kurses bauen aufeinander auf. Versuche daher die Aufgaben zeitnah zu bearbeiten.
2. Alle abzugebenden Quelltexte müssen ohne Warnungen und Fehler auf Deinem Rechner mit dem Befehl `clang -std=c11 -Wall -g` kompilieren.
3. Die Abgabe für den Quellcode erfolgt ausschließlich über unser Git im entsprechenden Branch. Nur wenn ein Ergebnis im [ISIS-Kurs](#) angezeigt wird, ist sichergestellt, dass die Abgabe erfolgt ist. Die Abgabe ist bestanden, wenn Du an Deinem Test einen grünen Haken siehst.
4. Du kannst bis zur Abgabefrist beliebig oft neue Versionen abgeben. Lies Dir die Hinweise der Tests genau durch, denn diese helfen Dir, Deine Abgabe zu korrigieren.  
**Bitte beachte, dass ausschließlich die letzte Abgabe gewertet wird.**
5. Die Abgabe erfolgt, sofern nicht anders angegeben, in folgendem Branch: `ckurs-b<xx>-a<yy>`, wobei `<xx>` durch die zweistellige Nummer des Aufgabenblattes und `<yy>` durch die entsprechende Nummer der Aufgabe zu ersetzen sind.
6. Gib für jede Aufgabe die Quellcodedatei(en) gemäß der Vorgabe ab. Im [ISIS-Kurs](#) werden zum Teil Vorgabedateien bereitgestellt. Nutze diese zur Lösung der Aufgaben.
7. Die Abgabefristen werden vom Server überwacht. Versuche, Deine Abgabe so früh wie möglich zu bearbeiten. Du minimierst so auch das Risiko, die Abgabefrist auf Grund von „technischen Schwierigkeiten“ zu versäumen. Eine Programmieraufgabe gilt als bestanden, wenn alle bewerteten Teilaufgaben bestanden sind.

### Aufgabe 1 Array-Aggregation

In dieser Aufgabe sollst Du Funktionen implementieren, die aus einer Zahlenreihe Werte berechnen. Wie in der Vorgabe zu dieser Aufgabe zu sehen, ist die Zahlenreihe in einem Array gespeichert.

#### Ausgabe eines Array

Implementiere als Erstes eine Funktion, die die Zahlen des Arrays jeweils durch ein Komma getrennt ausgibt. Nenne sie `print_array`. Verwende eine Schleife, um alle Elemente des Arrays zu erreichen.

Die Ausgabe erfolgt via `printf` auf der Kommandozeile. Achte darauf, nur auf Elemente zuzugreifen, die tatsächlich im Array vorhanden sind. Die Funktion `print_array` erhält als erstes Argument das Array und als zweites Argument die Länge des Arrays. Diese Funktion hat keinen Rückgabewert, sondern gibt mittels `printf` eine Zeile aus, die mit `Array:` beginnt.

#### Min & Max

Implementiere zwei Funktionen `min` und `max`. Die erste (`min`) soll das kleinste und die zweite (`max`) das größte Element des Arrays finden. Verwende die gleichen Parameter wie bei der Funktion `print_array`. Behalte dabei auch die Reihenfolge der Argumente bei. Beide Funktionen benutzen kein `printf`, sondern geben einen `int` Wert zurück. Erweitere die Funktion `main`, sodass Minimum und Maximum unter der Verwendung Deiner Funktionen ausgegeben werden. Beachte, dass die ausgegebenen Zeilen mit `Minimum:` bzw. `Maximum:` beginnen müssen.

#### Summe

Schreibe eine Funktion `sum`, die die Summe des Arrays berechnet. Dafür wird der Funktion `sum` als erstes Argument das Array, als zweites Argument die Länge des Arrays und als letztes Argument ein Pointer auf einen Integer (Call-by-Reference) mitgegeben. Diese Funktion soll ebenfalls kein `printf` verwenden und hat keinen Rückgabewert. Erweitere anschließend Deine `main` Funktion, sodass die Summe unter der Verwendung der soeben geschriebenen Funktion ausgegeben wird. Die Ausgabe muss mit `Summe:` beginnen.

Nutze die Funktionen, um nacheinander folgendes auszugeben:

1. das Array
2. das Minimum
3. das Maximum
4. die Summe

Ein Beispielaufruf ist in Listing 1 gezeigt.

Listing 1: Programmbeispiel

```
1 > clang -std=c11 -Wall -g ckurs_blatt05_aufgabe01.c -o ckurs_blatt05_aufgabe01
2 > ./ckurs_blatt05_aufgabe01
3 Array: 9, 4, 7, 8, 10, 5, 1, 6, 3, 2
4 Minimum: 1
5 Maximum: 10
6 Summe: 55
```

Um die Hausaufgabe zu vereinfachen, bitten wir Dich, die vorgegebene Programmstruktur zu verwenden (siehe Listing 2). Der Fall, in dem das Array leer ist, muss nicht behandelt werden. Die Abgabe muss folgenden Kriterien entsprechen:

- Das Array `int array[]`; wird zum Einlesen verwendet.
- Die Länge des Arrays ist in der Variable `int len`; gespeichert.
- Die Variablennamen `array` und `len` werden nicht in anderen Funktionen verwendet. (Sonst können wir die Aufgabe nicht testen.)
- Die Reihenfolge und die Anzahl der Parameter entspricht der Beschreibung auf diesem Aufgabenblatt.

- 
- Es werden vier Zeilen ausgegeben, wie in Listing 1 dargestellt. Vor den Ergebnissen wird der entsprechende String ausgegeben ("Array:", "Minimum:", "Maximum:", "Summe:").
  - Es werden nur ganze Zahlen ausgegeben.
  - Es dürfen keine weiteren Leerzeilen ausgegeben werden.

#### Listing 2: Mögliche Programmstruktur

---

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void print_array(int array[], int len) {
5     // HIER Code einfügen
6 }
7
8 // Schreibe die Funktion "sum", "min" und "max"
9
10 int main() {
11     int array[] = {9, 4, 7, 8, 10, 5, 1, 6, 3, 2};
12     int len = 10;
13     print_array(array, len);
14     // Gebe hier nacheinander das Minimum, Maximum und die Summe
15     // aus. Trenne die Werte durch einen einzelnen Zeilenumbruch.
16 }
```

---

Nutze zur Lösung der Aufgabe die Vorgaben aus unserem [ISIS-Kurs](#). Füge Deine Lösung als Datei `ckurs_blat05_aufgabe01.c` im entsprechenden Abgabebereich in Dein persönliches Repository ein und übertrage die Lösung an die Abgabepattform.