

*Frankfurt University of Applied Sciences  
Fachbereich 2: Informatik und Ingenieurwissenschaften*

**EINFÜHRUNG IN DIE PROGRAMMIERUNG  
MIT DER PROGRAMMIERSPRACHE C  
(EifP, WS2025/26)**

Prof. Dr. Thomas Gabel  
Roman Ahmad, Emre Özöner, Gina Romanazzi

**Aufgabenblatt 10**

**Aufgabe 47: Das kleinste gemeinsame Vielfache**

**Inhalte und Sprachkonzepte der vorliegenden Aufgabe**

- Algorithmusentwurf
- programmiertechnische Umsetzung eines einfachen Algorithmusentwurfs
- Schleifen, Operatoren

**Aufgabenstellung**

- Geben Sie einen Algorithmus zur Bestimmung des kleinsten gemeinsamen Vielfachen (kgV) zweier natürlicher Zahlen  $m$  und  $n$  in Pseudocode-Schreibweise an. Verwenden Sie dabei die aus der Vorlesung bekannten Kontrollstrukturen (Fallunterscheidung, Schleifen).  
*Hinweis: Es gibt mehrere Ansätze zur Ermittlung des kgV. Sie dürfen gern den für Sie einfachsten und intuitivsten Ansatz umsetzen.*  
*Hinweis: Eine Möglichkeit zur kgV-Bestimmung besteht darin, die Primzahlzerlegung von  $m$  und  $n$  zu benutzen. Beispiel:  $90 = 2 \cdot 3 \cdot 3 \cdot 5$ ,  $75 = 3 \cdot 5 \cdot 5$ . Damit  $\text{kgV}(75, 90) = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 5 = 450$*
- Implementieren Sie in a) entworfenen Algorithmus in der Programmiersprache C. Vom Nutzer sollen also zunächst die beiden natürlichen Zahlen eingehen, dann das kgV ermittelt und schließlich dieses ausgegeben werden.  
*Hinweis: Es gibt mehrere Ansätze zur Ermittlung des kgV. Sie dürfen gern den für Sie einfachsten und intuitivsten Ansatz umsetzen.*  
*Hinweis: Eine Möglichkeit zur kgV-Bestimmung besteht darin, die Primzahlzerlegung von  $m$  und  $n$  zu benutzen. Beispiel:  $90 = 2 \cdot 3 \cdot 3 \cdot 5$ ,  $75 = 3 \cdot 5 \cdot 5$ . Damit  $\text{kgV}(75, 90) = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 5 = 450$*
- Ermitteln Sie mit Ihrem Programm  $\text{kgV}(89, 233)$ ,  $\text{kgV}(442, 715)$  sowie  $\text{kgV}(943, 1271)$ .

**Aufgabe 48: Zeichenkettenüberprüfungen**

**Inhalte und Sprachkonzepte der vorliegenden Aufgabe**

- Einlesen von Zeichenketten

- Vergleiche von Zeichenketten

### Aufgabenstellung

Schreiben Sie ein C-Funktion, welche bestimmt, ob zwei von der Tastatur eingelesene Zeichenketten gleich sind. Wenn das nicht der Fall ist, so soll überprüft werden, ob und wie oft eine der Zeichenketten in der anderen enthalten ist.

Beispiel: "abc" ist in der Zeichenkette "abcdjfjabcldghreobcdhls" drei Mal enthalten.

Ihre Lösung soll ohne Verwendung der in `string.h` deklarierten Funktionen auskommen.

### Aufgabe 49: Einfachste Zeigernutzung

#### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Zeiger

### Aufgabenstellung

Erweitern Sie Ihr Programm aus der Aufgabe "Grundrechenarten" folgendermaßen:

- Führen Sie zwei Zeigervariablen `i` und `j` ein und weisen Sie diesen die Adressen der Variablen `a` und `b` zu.
- Vertauschen Sie nun die Inhalte der beiden Zeiger, d.h. `i` und `j` sollen die Adressen tauschen, auf die sie zeigen.
- Führen Sie die Operationen aus der Aufgabe "Grundrechenarten" unter Verwendung der Zeigervariablen `i` und `j` durch. Die Ergebnisse sollen in der gleichen Art und Weise wie in der Aufgabe "Grundrechenarten" ausgegeben werden.

### Aufgabe 50: Umfang und Radius

#### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Zeiger initialisieren
- Zeiger als Übergabeparameter von Funktionen (Call by Reference)

### Aufgabenstellung

In der folgenden Aufgabe ist ein Programm `kreis.c` zu schreiben, das die Fläche und den Umfang eines Kreises berechnet. Das Einlesen vom Radius und die Berechnungen sollen ausschließlich in Funktionen stattfinden.

Deklarieren Sie 3 positive reelle Variablen für Radius, Fläche und Umfang im Hauptprogramm sowie zwei Zeiger vom entsprechenden Typ: einen für den Radius und einen für Fläche bzw. Umfang, welche Sie als Übergabeparameter der Funktionen benutzt werden. Diese Funktionen sollen folgende Prototypen haben:

- `void read_radius(double* radius);`
- `void circle_circumference(double radius, double* circumference);`
- `void circle_area(double radius, double* area);`

Für  $\pi$  ist eine globale Konstante vom Typ `double` zu verwenden.

*Hinweis:* Für den Umfang und für die Fläche steht nur ein Zeiger zur Verfügung, und so muss dieser vor jeder Berechnung entsprechend initialisiert werden.

*Zusatzaufgabe:* Modifizieren Sie Ihr Programm nun so, dass Umfang und Fläche des Kreises mittels eines einzigen Funktionsaufrufes der Funktion `void circle_circumference_and_area(double radius, double* area, double* circumference);` berechnet werden, wobei Sie die Adressen Ihrer im Hauptprogramm deklarierten Variablen für Fläche und Umfang übergeben.

## Aufgabe 51: Dynamische Speicherallokation

### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Grundlagen dynamischer Speicherallokation

### Aufgabenstellung

Schreiben Sie ein C-Programm `allocator.c`, das vom Benutzer eine positive natürliche Zahl  $n$  einliest und anschließend zwei dynamisch allokierte Felder der Größe  $n$  erstellt.

- a) Das erste Feld soll Integer-Werte speichern können und mit den Quadraten der Werte von 0 bis  $n - 1$  befüllt werden.
- b) Das zweite Feld soll Double-Werte speichern können und mit den Quadratwurzeln der Zahlen von 0 bis  $n - 1$  befüllt werden.
- c) Geben Sie die Inhalte beider Felder rückwärtig in gut lesbarem Format aus, d.h. beginnen Sie Ihre Ausgabe am Ende der Felder (bei den Einträgen für  $n - 1$ ).

## Aufgabe 52: Random Walk

### Inhalte und Sprachkonzepte der vorliegenden Aufgabe

- Nutzung der Bibliothek `advanced_console_output.h`
- Random Walk (Zufallsbewegung)

### Aufgabenstellung

Laut Wikipedia ist ein Random Walk (deutsch: zufällige stochastische Irrfahrt, zufällige Schrittfolge, Zufallsbewegung oder Zufallsweg) ein mathematisches Modell für eine Verkettung zufälliger Bewegungen. Schreiben Sie unter Benutzung der Bibliothek *Advanced Console Output* ein C-Programm, das den folgenden Anforderungen genügt:

- Der Zufallszahlengenerator soll systemzeitbasiert initialisiert werden.
- Der Bildschirm soll gelöscht werden.
- Der Cursor soll in der Mitte des Bildschirms platziert werden.
- In jedem Zeitschritt soll an der aktuellen Stelle des Cursors das Zeichen '\*' ausgegeben werden.
- Die Schriftfarbe für das ausgegebene '\*'-Zeichen soll in jedem Schritt heller werden. Die Farbe selbst darf nach eigenem Belieben gewählt werden.
- In jedem Zeitschritt soll der Cursor zufällig um eine Position nach links, rechts, oben oder unten versetzt werden. Dies soll unter Verwendung der Auswahlanweisung realisiert werden.
- Nach jedem Zeitschritt sollen die Ausgabe "geflusht" und die Ausführung des Programms für 100 Millisekunden gewartet werden.
- Das Programm soll für maximal 256 Zeitschritte ausgeführt werden oder aber, bis der Cursor einen der Bildschirmränder erreicht.
- Unmittelbar vor Programmende ist der Cursor an den Anfang der untersten Zeile zu versetzen.
- Lassen Sie sich von dem in campUAS unter `U10_Resources.tar.gz` bereitgestellten Beispielprogramm inspirieren.