



---

## Freiwilliges Programmieren 1

---

In diesem Freiwilligen Programmieren werden Wiederholungsaufgaben zu den Themen 2 und 3 des Vorkurses behandelt. Für jede Teilaufgabe soll ein C-Programm erstellt und dazu die in Worten beschriebenen Anweisungen in C-Anweisungen umformuliert werden. Dabei ist jeder Satz in genau eine C-Anweisung zu überführen.

Erstellen Sie für jede Teilaufgabe jeweils eine C-Datei mit einer eigenen `main`-Funktion. Kompilieren Sie Ihre Programme mit den Compilerschaltern `-ansi` `-pedantic` `-Wall` `-Wextra` und führen Sie sie aus (jeweils über ein Kommandozeilen-Programm).

### Aufgabe 1.1 (*Ganze Zahlen*)

a)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert 4711 zu.
- Geben Sie den Wert von `n` aus.

b)

- Deklarieren Sie eine `int`-Variable `n`.
- Weisen Sie `n` den Wert `-INT_MAX` zu.
- Geben Sie den Wert von `n` aus.

c)

- Deklarieren Sie eine `int`-Variable `n`.
- Erzeugen Sie mit `rand` eine ganze Zufallszahl zwischen 10 und 20 (jeweils eingeschlossen) und weisen Sie `n` den Wert der Zufallszahl zu
- Geben Sie den Wert von `n` aus.

### Aufgabe 1.2 (*Arithmetische Rechenausdrücke*)

a)

- Deklarieren Sie drei `int`-Variablen `a`, `b` und `c`.

- Weisen Sie **a** den Wert 12 zu.
- Weisen Sie **b** den Wert 21 zu.
- Weisen Sie **c** den Wert der Addition von **a** und **b** zu.
- Geben Sie den Wert von **c** aus.

b)

- Deklarieren Sie drei **int**-Variablen **a**, **b** und **c**.
- Weisen Sie **a** den Wert 35 zu.
- Weisen Sie **b** den Wert 9 zu.
- Weisen Sie **c** den Wert des Rests der (ganzzahligen) Division von **a** und **b** zu.
- Geben Sie den Wert von **c** aus.

c)

- Deklarieren Sie eine **int**-Variable **d**.
- Weisen Sie **d** den Wert 22 zu.
- Verdoppeln Sie dann den Wert von **d**.
- Geben Sie den Wert von **d** aus.

### Aufgabe 1.3 (*Auswertungsreihenfolge und Overflow*)

a)

- Deklarieren Sie drei **int**-Variablen **a**, **b** und **c** und weisen Sie **a** den Wert 8, **b** den Wert 7 und **c** den Wert 6 zu.
- Geben Sie den Wert der folgenden Rechnung aus: Addieren Sie zuerst **a** und **b**, und multiplizieren Sie dann das Ergebnis mit **c**.

b)

- Deklarieren Sie eine **int**-Variable **d**.
- Weisen Sie ihr den Wert **INT\_MIN** zu.
- Erniedrigen Sie dann den Wert von **d** um 1.
- Geben Sie den Wert von **d** aus.

c)

- Deklarieren Sie eine **int**-Variable **d**.
- Weisen Sie ihr den Wert **INT\_MAX** zu.
- Addieren Sie dann **INT\_MAX** auf den Wert von **d** und weisen Sie das Ergebnis wieder **d** zu.
- Geben Sie den Wert von **d** aus.

### Aufgabe 1.4 (*ASCII-Zeichen*)

a)

- Deklarieren Sie eine **char**-Variable **e**.

- Weisen Sie ihr den Wert `'\"'` zu.
  - Geben Sie den Wert von `e` als Zeichen aus.
  - Geben Sie den Wert von `e` als ganze Zahl aus.
- b)
- Geben Sie das ASCII-Zeichen mit dem ASCII-Code 64 aus (ohne in der ASCII-Tabelle nachzusehen).
- c)
- Erzeugen Sie mit einem `printf`-Aufruf die folgende Ausgabe:
- ```

*   *
 *  *
*   *
 *  *
*   *

```

### Aufgabe 1.5 (*Zeichenfunktionen*)

- a) Geben Sie jeweils in einer eigenen Zeile den Rückgabewert des Funktionsaufrufs `isupper(c)` für verschiedene Zeichen `c` aus:
- Für das Zeichen `'v'`
  - Für das Zeichen `'W'`
  - Für das Zeichen `'?'`
  - Für das Zeichen `'1'`
  - Für das Zeichen `'\v'`
- b) Geben Sie jeweils in einer eigenen Zeile den Rückgabewert des Funktionsaufrufs `toupper(c)` für verschiedene Zeichen `c` als Zeichen zwischen zwei Hochkommas (z. B. `'*'`) aus:
- Für das Zeichen `'v'`
  - Für das Zeichen `'W'`
  - Für das Zeichen `'?'`
  - Für das Zeichen `'1'`
  - Für das Zeichen `'\v'`
- c) Geben Sie jeweils in einer eigenen Zeile den Rückgabewert des Funktionsaufrufs `isalnum(c)` für verschiedene Zeichen `c` aus:
- Für das Zeichen `'v'`
  - Für das Zeichen `'W'`
  - Für das Zeichen `'?'`
  - Für das Zeichen `'1'`
  - Für das Zeichen `'\v'`