



---

## Freiwilliges Programmieren 3

---

In diesem Freiwilligen Programmieren werden Wiederholungsaufgaben zu den Themen 4 und 5 des Vorkurses behandelt.

Erstellen Sie für jede Teilaufgabe jeweils eine C-Datei mit einer `main`-Funktion. Achten Sie bei Programmieraufgaben auf *Kompilierbarkeit* und *Einhaltung der Coding Conventions*; Kompilieren Sie Ihre Programme mit den Compilerschaltern `-ansi` `-pedantic` `-Wall` `-Wextra`. Achten Sie darauf, dass trotz Verwendung dieser Schalter keine Fehler-/Warnmeldungen erzeugt werden. Führen Sie Ihre Programme jeweils aus.

### Aufgabe 3.1 (*Verschachtelte Schleifen*)

a)

- Geben Sie mit zwei ineinander verschachtelten `for`-Schleifen ein Lottofeld mit sieben Zeilen und sieben Spalten aus (also eine Tabelle mit den Zahlen von 1 bis 49).

Die Tabelle soll dieses Aussehen haben:

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

b)

- Geben Sie mit zwei ineinander verschachtelten `for`-Schleifen 10 Zeilen aus (äußere Schleife), wobei die  $i$ -te Zeile aus  $10 - i$  Leerzeichen gefolgt von  $i$  `'*'`-Zeichen bestehen (innere Schleife), indem Sie pro Schleifendurchlauf der inneren Schleife abhängig vom aktuellen Wert  $i$  entweder ein Leerzeichen oder das Zeichen `'*'` ausgeben.

Die Ausgabe sieht so aus:

```
*
**
***
****
```

```

*****
*****
*****
*****
*****
*****
*****

```

### Aufgabe 3.2 (*Einfache Funktionen*)

Erstellen Sie für jede Teilaufgabe jeweils eine C-Datei mit einer `main`-Funktion und einer weiteren Funktion gemäß der Aufgabenstellung. In der `main`-Funktion testen Sie jeweils die andere Funktion.

a)

Erstellen Sie eine Funktion mit dem Prototyp `double decrement(double x)` nach folgenden Vorgaben:

- Die übergebene Dezimalzahl soll um 1 erniedrigt und das Ergebnis zurückgegeben werden.

In der `main`-Funktion testen Sie die `decrement`-Funktion nach folgenden Vorgaben:

- Erzeugen Sie eine zufällige Dezimalzahl `d` aus dem Bereich von 0.5 bis 9.5.
- Geben Sie `d` aus.
- Geben Sie `decrement(d)` aus.

b)

Erstellen Sie eine Funktion mit dem Prototyp `int my_toupper(int c)` nach folgenden Vorgaben:

- Ist der übergebene Wert der ASCII-Code eines lateinischen Kleinbuchstabens, soll der ASCII-Code des zugehörigen Großbuchstabens zurückgegeben werden.
- Ist der übergebene Wert nicht der ASCII-Code eines lateinischen Kleinbuchstabens, soll `c` zurückgegeben werden.
- Die Verwendung von Funktionen aus `ctype.h` ist nicht erlaubt!

In der `main`-Funktion testen Sie die `my_toupper`-Funktion nach folgenden Vorgaben:

- Erzeugen Sie eine zufällige ganze Zahl `c` zwischen 0 und 127 (jeweils einschließlich).
- Geben Sie `c` als ASCII-Zeichen aus.
- Geben Sie `my_toupper(c)` als ASCII-Zeichen aus.

### Aufgabe 3.3 (*Fortgeschrittene Funktionen*)

Erstellen Sie für jede Teilaufgabe jeweils eine C-Datei mit einer `main`-Funktion und einer weiteren Funktion gemäß der Aufgabenstellung. Die Erstellung der Funktion erfordert jeweils die Benutzung von geeigneten Schleife. In der `main`-Funktion testen Sie jeweils die andere Funktion.

a)

Erstellen Sie eine Funktion mit dem Prototyp `void print_alarm(int size)` nach folgenden Vorgaben:

- Falls `size < 0` gilt, soll mit einer Fehlerausgabe abgebrochen werden.
- Es sollen `size` Alarmsignale (Systemtöne bzw. Beeps) ausgegeben werden.

In der `main`-Funktion testen Sie die `print_alarm`-Funktion nach folgenden Vorgaben:

- Rufen Sie die `print_alarm`-Funktion für den Übergabewert 3 auf.
- Rufen Sie die `print_alarm`-Funktion für den Übergabewert 0 auf.
- Rufen Sie die `print_alarm`-Funktion für den Übergabewert -5 auf.
- Hinweis: Es kann sein, dass Ihr PC das Programm zu schnell ausführt und die korrekte Anzahl an Alarmtönen nicht ausgegeben wird - in diesem Fall verwenden Sie direkt nach dem Aufruf für die Tonausgabe die Anweisung `getchar()` und das Programm pausiert solange bis vom Benutzer die Enter-Taste betätigt wird.

b)

Erstellen Sie eine Funktion mit dem Prototyp `double power3(int n)` nach folgenden Vorgaben:

- Es soll der  $3^n$  berechnet und als `double`-Wert zurückgegeben werden (siehe auch Freiwilliges Programmieren 2).
- Die Verwendung von Funktionen aus `math.h` ist nicht erlaubt!

In der `main`-Funktion testen Sie die `power3`-Funktion nach folgenden Vorgaben:

- Berechnen Sie in einer Schleife die Werte  $3^n$  der Zahlen von 1 bis 13 und geben Sie diese zeilenweise aus.

### Aufgabe 3.4 (Felder)

Erstellen Sie für jede Teilaufgabe jeweils eine C-Datei mit einer `main`-Funktion (ohne weitere Funktionen).

a)

- Deklarieren Sie `int`-Feld `a` mit 4 Komponenten.
- Speichern Sie in der ersten Komponente den Wert 144.
- Speichern Sie in der zweiten Komponente die Quadratwurzel des Werts der ersten Komponente.
- Ziehen Sie die zweite Komponente von der ersten Komponente ab und speichern Sie das Ergebnis in der dritten Komponente.
- Speichern Sie in der vierten Komponente die Division der dritten Komponente durch 6
- Geben Sie zeilenweise die Werte der Komponenten von `a` aus (Schleife benutzen!).

b)

- Deklarieren Sie `char`-Feld `b` mit 16 Komponenten.
- Speichern Sie in den Komponenten von `b` die Ziffern von '0' bis '9' und die Großbuchstaben von 'A' bis 'F' (Schleife benutzen!).
- Geben Sie hintereinander ohne Leerzeichen die Werte der 13-ten, der 1-ten, der 14-ten und der 15-ten Komponente als Zeichen aus.

### Aufgabe 3.5 (Funktionen für Felder)

Erstellen Sie für alle Teilaufgaben zusammen eine C-Datei mit einer `main`-Funktion und ergänzen Sie in jeder Teilaufgabe eine weitere Funktion gemäß der Aufgabenstellung. Testen Sie anschließend in der `main`-Funktion die neuen Funktionen (dabei können Sie sich an der `main`-Funktion

aus den Vorlesungsfolien orientieren).

a)

Erstellen Sie eine Funktion mit dem Prototyp `void array_randneg(int a[], int size)` nach folgenden Vorgaben:

- Es soll in die ersten `size` Komponenten des Feldes `a` jeweils ein negativer Zufallswert gespeichert werden.

b)

Erstellen Sie eine Funktion mit dem Prototyp `int array_isneg(int a[], int size)` nach folgenden Vorgaben:

- Falls die ersten `size` Komponenten des Feldes `a` jeweils negative Zahlen sind, soll 1 zurückgegeben werden, sonst soll 0 zurückgegeben werden.