



# Praktikum Informatik II – BMT

## Programmierung in C

### Praktikum 5 – Funktionen und Arrays

Prof. Dr. Dieter Koller

# Verständnisfragen zu Funktionen

---

Was ist falsch an den Definitionen folgender Funktionen:

- a) 

```
void func(int a)
{ return a*a; }
```
- b) 

```
double func(double x)
{ printf("Testwert: %lf\n", x); }
```
- c) 

```
double func(double x, y)
{ return x*y; }
```
- d) 

```
int func(int a)
{ return a*a };
```

# Aufgabe 1: Mathe-Funktion Logarithmus

---

1. Schreiben Sie ein C-Programm mit einer Funktion, die den Logarithmus einer Zahl  $x$  zu einer beliebigen Basis  $b$  berechnet nach der Formel

$$\log_b(x) = \frac{\log_{10}(x)}{\log_{10}(b)} = \frac{\ln(x)}{\ln(b)}$$

Die **log10(x)**-Funktion (=  $\log_{10}(x)$  : Logarithmus zur Basis 10) und auch **log(x)** (=  $\ln(x)$ : Logarithmus zur Basis  $e$ ) sind in der Bibliothek `<math.h>` enthalten. Welchen Rückgabetyt brauchen Sie? Welche Parameter mit welchen Datentypen bieten sich an?

2. Schreiben Sie eine weitere Funktion, die einen übergebenen Text ausgibt, dann eine **positive Zahl > 0** einliest und diese zurückgibt. Bei Fehleingaben wird in einer Schleife die Eingabe wiederholt.
3. Schreiben Sie nun das Hauptprogramm so, dass nach einem positiven Wert und nach einer Basis gefragt wird. Die Meldungen werden zur Ausgabe der Funktion übergeben. Schließlich berechnen Sie den Logarithmus und geben diesen mit einem entsprechenden Text aus.

## Aufgabe 2: Funktion – Test auf Gerade Zahl

---

Schreiben Sie ein Programm in C, mit dem Sie zunächst eine **ganze Zahl** eingeben können. Benutzen Sie dann eine Funktion, die Sie selbst schreiben:

```
int iSEven(int);
```

mit der Sie dann überprüfen können, ob die eingegebene Zahl gerade ist. Diese Funktion liefert **1** (true) zurück, wenn die Zahl gerade ist, ansonsten **0** (false). Geben Sie dann das Ergebnis wie im Beispiel aus.

```
C:\src\Info2\p5>p5_iseven
Geben Sie eine ganze Zahl ein: 12
Die Zahl 12 ist gerade

C:\src\Info2\p5>p5_iseven
Geben Sie eine ganze Zahl ein: 13
Die Zahl 13 ist ungerade
```

# Aufgabe 3: Zeichen einlesen und einschränken

---

1. Schreiben Sie ein C Programm, das mit einer Funktion

`char readChar()`

nur Buchstaben (a-z, A-Z), Leerzeichen, und die Eingabetaste (ASCII 13) zulässt. Ein gültiges Zeichen wird zurückgegeben, ein ungültiges Zeichen löst einen Ton aus (ASCII 7) und wiederholt die Eingabe (Schleife).

Hinweis: Benutzen Sie die `getch()` Funktion, um ein Zeichen einzulesen.

2. Erweitern Sie das obige Programm durch eine weitere Funktion

`int readLine(char *text, int n)`

die die `readChar()`-Funktion benutzt und die Eingabe auf n Zeichen beschränkt. Die Eingabetaste beendet die Eingabe.

In der Hauptfunktion lesen Sie mit Ihrer Funktion einen Satz ein und geben ihn dann wieder aus.

# Aufgabe 4: Eindimensionale Arrays – Statistik

---

Schreiben Sie ein Programm, das Sie auffordert, fünf ganze Zahlen einzugeben. Lesen Sie die fünf Zahlen in ein Array ein. (Nutzen Sie ein `#define` für die Arraygröße!)

Berechnen Sie dann den Maximalwert, den Mittelwert und die Summe der fünf Zahlen und geben diese aus.

# Aufgabe 5: Eindimensionale Arrays – Tageszähler

---

Schreiben Sie ein Programm, das zur Eingabe eines Datums, d.h. Tag, Monat und Jahr fragt. Berechnen Sie dann, der wievielte Tag im Jahr das ist, d.h. wie viele Tage seit Jahresbeginn vergangen sind.

Schreiben Sie dazu eine eigene Funktion:

```
int istSchaltjahr(int j);
```

mit Hilfe des Modulo Operators. Diese gibt wahr (1) zurück, wenn j ein Schaltjahr ist ansonsten falsch (0). D.h. es ist wahr, wenn j durch 4, nicht durch 100, aber wieder durch 400 teilbar ist.

Z.B. 2000 war ein Schaltjahr (durch 4, 100 und 400 teilbar).

Legen Sie die Anzahl der Tage pro Monat in einem Array fest und summieren Sie dann die Tage seit dem 1.1. des jeweiligen Jahres auf!

**Zusatzaufgabe:** Überprüfen Sie auch, ob die Eingaben plausibel sind.

# Aufgabe 6: 2D Vektoren

---

Schreiben Sie ein Programm, das Sie zur Eingabe eines 2D-Vektors auffordert (vx, vy).

Berechnen Sie dann die Länge dieses Vektors mit Hilfe einer eigenen Funktion, die die Quadratwurzelfunktion `sqrt()` aus der Bibliothek `math.h` benutzt. (Der Funktionsprototyp von `sqrt()` ist `double sqrt(double)`).

Lassen Sie dann einen zweiten Vektor eingeben und berechnen Sie den Summenvektor und auch dessen Länge.

**Frage:** Was müsste man ändern, um Vektoren eine beliebigen Dimension n zu eingeben und zu bearbeiten?