

**Ausgabe:** Donnerstag, 14.10.2009

**Abgabe:** Sonntag, 25.10.2008

**Abgabe ausschließlich per Mail an den zuständigen Übungsgruppenbetreuer**

**Wichtig: Benennen Sie die Dateien wie in der jeweiligen Aufgabe angegeben!**

### 1.1 Datentypen

(aufgabe1\_1.c, 1 Punkte)

Hochsprachen wie C und C++ stellen eine Reihe von Datentypen für die Verarbeitung von Zahlen bereit. Diese unterscheiden sich durch die Darstellung eines numerischen Wertes, sowie durch die Genauigkeit. Die Funktion `sizeof()` gibt an, welcher Speicherplatz für eine Variable benötigt wird. Schreiben Sie ein Programm, das für jeweils vier verschiedene Datentypen den benötigten Speicherplatzbedarf in Byte an die Standardausgabe schreibt.

### 1.2 Maschinengenauigkeit

(aufgabe1\_2.c, 2 Punkte)

Die Maschinengenauigkeit `eps` kann über den folgenden Pseudo-Code bestimmt werden:

1. `eps = 1`  
   `alpha = 1`
2. Halbiere `eps`.
3. Vergleiche `alpha` mit `alpha+eps`.  
   Falls beide Werte identisch sind, verdopple `eps`, um die gesuchte Maschinengenauigkeit zu erhalten.  
   Andernfalls wiederhole die Vorschrift beginnend bei Schritt 2.

Schreiben Sie ein Programm zur Bestimmung der Maschinengenauigkeit. Welche Werte ergeben sich für (i) `alpha = 1`, (ii) `alpha = 10-4` und (iii) `alpha = 1011`? Was genau beschreibt die Maschinengenauigkeit und warum wäre es falsch, in Schritt 2 den Wert von `eps` zu dritteln?

### 1.3 Arrays und Strings

(aufgabe1\_3.c, 3 Punkte)

Mehrere Variablen gleichen Typs lassen sich in sogenannten Arrays zusammenfassen. Der C-Ausdruck

```
float x[2];  
x[0]=1.0;  
x[1]=0.5;
```

z.B. erstellt ein Array (oder Vektor) aus zwei Gleitkommazahlen. Ähnlich verhält es sich mit Buchstabenfolgen. Hier wird der Ausdruck "Hallo, Welt" in einer Variable gespeichert:

```
char text[] = "Hallo, Welt";
```

Machen Sie sich mit dem Konzept von Arrays/Vektoren/Feldern vertraut und schreiben Sie ein Programm, das einen Text mit maximal 80 Zeichen mittels der Funktion `scanf()` von `STDIN` einliest und dann rückwärts nach `STDOUT` schreibt.<sup>1</sup>

<sup>1</sup>`STDIN` und `STDOUT` bezeichnen die Standardein- bzw -ausgabe.

### 1.4 Mittelwert und Fehler

(aufgabe1\_4.c, 3 Punkte)

Sei  $\{x_1, \dots, x_N\}$  eine Stichprobe vom Umfang  $N$  – z.B. Messergebnisse eines physikalischen Experiments. Dann ist durch

$$\langle x \rangle := \frac{1}{N} \sum_{n=1}^N x_n \quad (1)$$

das *arithmetische Mittel* und durch

$$\sigma := \sqrt{\frac{1}{N-1} \sum_{n=1}^N (x_n - \langle x \rangle)^2} \quad (2)$$

ein Schätzer für die *Standardabweichung* gegeben. Schreiben Sie nun ein Programm, das mit Hilfe der Funktion `scanf()` eine Stichprobe von fünf Zahlen von der Standardeingabe einliest und  $\langle x \rangle$ , sowie  $\sigma$  ausgibt.

### 1.5 Fehler in der endlichen Exponentialfunktion

(aufgabe1\_5.c, 3 Punkte)

Transzendente Funktionen wie die Exponentialfunktion `exp()` oder die trigonometrischen Funktionen `sin()`, `cos()`, `tan()` sind durch die arithmetischen Möglichkeiten eines Prozessors nicht direkt zugänglich, sondern müssen auf eine polynomielle Darstellung abgebildet werden. Dazu wird oft die Formel von Taylor verwendet:

$$f(x) \approx \sum_{n=0}^N \frac{1}{n!} \left. \frac{d^n f}{dx^n} \right|_a (x-a)^n \quad (3)$$

Implementieren Sie die Taylorentwicklung der Exponentialfunktion um den Entwicklungspunkt  $a = 0$  bis zum Grad  $N = 5$  und bestimmen Sie für  $x \in \{0, 1, \dots, 10\}$  den relativen Fehler gegenüber der in `math.h` implementierten Funktion `exp()`.