

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 4

Aufgabe 4.1*. Für einen gegebenen Vektor $v \in \mathbb{R}^n$ sind *arithmetisches* und *harmonisches* Mittel definiert durch

$$x_{\text{arit}} := \frac{1}{n} \sum_{i=1}^n x_i,$$
$$x_{\text{harm}} := \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}.$$

Schreiben Sie eine void Funktion `means`, die zu einem gegebenen Vektor sowohl arithmetisches, als auch harmonisches Mittel berechnet. Achten Sie darauf, dass Sie hier mehrere Rückgabewerte haben. Wie können Sie das realisieren? Speichern Sie den Source-Code unter `means.c` in das Verzeichnis `serie04`.

Aufgabe 4.2*. Was ist eine Schleife? Welche unterschiedlichen Typen von Schleifen gibt es? Erklären Sie die Unterschiede. Was ist der Unterschied und der Zusammenhang zwischen einer Variable und einem Pointer?

Aufgabe 4.3*. Welchen Output liefert das folgende Programm und warum?

```
#include <stdio.h>

void square(double* x)
{
    double* y;
    *y=(*x)*(*x);
}

main()
{
    double x=2.1;
    square(&x);
    printf("x^2=%f\n", x);
    return 0;
}
```

Verändern Sie das Programm geeignet, sodass der Output den Erwartungen entspricht. Speichern Sie den Source-Code unter `square.c` in das Verzeichnis `serie04`.

Aufgabe 4.4*. Das *babylonische Wurzelziehen* ist ein iteratives Verfahren zur Bestimmung der Quadratwurzel einer Zahl a . Ausgehend von einem beliebigen Startwert $x_0 > 0$ konvergiert die Folge

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

gegen \sqrt{a} . Implementieren Sie dieses Verfahren, wobei a und x_0 von der Tastatur einzulesen sind. Brechen Sie die Iteration ab, sobald $|x_{n+1} - x_n| \leq \tau$ für ein ebenfalls einzulesendes $\tau > 0$ gilt. Speichern Sie den Source-Code unter `rootmaster.c` in das Verzeichnis `serie04`.

Aufgabe 4.5. In der Programmierung besteht oft eine Notwendigkeit nach Zufallszahlen. In C können diese (Pseudo-) Zufallszahlen mit dem Befehl `rand()` realisiert werden. Finden Sie heraus, wie dieser Befehl zu verwenden ist. Schreiben Sie eine Funktion `newrand(min, max)`, die eine Zufallszahl aus dem Intervall $[\text{min}, \text{max}]$ zurückgibt. Ein aufrufendes Hauptprogramm soll nun so lange Zufallszahlen addieren, bis die Summe eine gewisse Schranke c , die zuvor von der Tastatur eingelesen wurde übersteigt. Speichern Sie den Source-Code unter `alleszufall.c` in das Verzeichnis `serie04`.

Aufgabe 4.6. Die Gleitkommaaddition ist nicht assoziativ. Aufgrund von nicht vermeidbaren Rundungsfehlern gilt für 3 reelle Zahlen a, b, c also im Allgemeinen

$$a + (b + c) \neq (a + b) + c.$$

Finden Sie hierfür ein Beispiel. Schreiben Sie ein kurzes Programm, das Ihre Behauptung verifiziert. Wie verhält es sich mit der Gleitkommamultiplikation? Gibt es auch noch weitere mathematische Gesetze, die von der Gleitkommaarithmetik verletzt werden?

Aufgabe 4.7. Schreiben Sie eine Funktion `findmax`, die das Maximum eines gegebenen Vektors $v \in \mathbb{R}^n$ findet und sowohl den Wert des Maximums als auch den ersten Index, an dem dieser Wert auftritt, zurückgibt. Der Vektor soll, wie zuvor, von der Tastatur eingelesen werden.

Aufgabe 4.8. Schreiben Sie eine Funktion `binomial`, die mittels *einer* geeigneten Schleife den Binomialkoeffizienten $\binom{n}{k}$ berechnet. Dazu realisiere man die gekürzte Form

$$\binom{n}{k} = \frac{n \cdot (n-1) \cdots (n-k+1)}{k \cdot (k-1) \cdots 1}.$$

Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem $k, n \in \mathbb{N}_0$ mit $k \leq n$ eingelesen werden und $\binom{n}{k}$ ausgegeben wird. Schreiben Sie weiters eine Lösung mit *zwei* Schleifen, bei der Zähler und Nenner getrennt berechnet werden. Speichern Sie den Source-Code unter `binomial.c` in das Verzeichnis `serie04`. Welche der zwei Implementierungen ist klüger und warum?

Aufgabe 4.9. Was ist die größte, was die kleinste Zahl, die von einer Variable des Typs `long` gespeichert werden kann? Schreiben Sie ein Programm, welches Die Bit-Größe von `long` berechnet und die beiden Zahlen ausgibt. Erzeugen Sie die gleiche Ausgabe schließlich für den Datentyp `unsigned int`. Speichern Sie den Source-Code unter `big.c` in das Verzeichnis `serie04`.

Aufgabe 4.10. Schreiben Sie eine `void` Funktion `swap`, welche die Werte von zwei Integern a und b vertauscht. Speichern Sie den Source-Code unter `swap.c` in das Verzeichnis `serie04`.