

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 2

Aufgabe 2.1. Bei einem aktuellen Smartphone-Tarif von A1 kostet eine Minute Telefonieren in ein beliebiges Mobilfunknetz nach Verbrauch der Freiminuten 0.29 €. Schreiben Sie ein Programm, das ein gewisses Guthaben $g \in \mathbb{R}$ von der Tastatur einliest und ausgibt wie lange (MIN) man damit telefonieren kann. Speichern Sie den Source-Code unter `a1.c` in das Verzeichnis `serie02`.

Aufgabe 2.2. Schreiben Sie eine Funktion `skalarprodukt`, die zu gegebenen Vektoren $\mathbf{u} = (a, b, c)^T$ und $\mathbf{v} = (x, y, z)^T$ das Skalarprodukt $w = \mathbf{u} \cdot \mathbf{v}$ mit $w = ax + by + cz$ berechnet. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Parameter a, b, c, x, y, z über die Tastatur eingelesen und das Ergebnis ausgegeben werden. Speichern Sie den Source-Code unter `skalarprodukt.c` in das Verzeichnis `serie02`.

Aufgabe 2.3. Schreiben Sie eine `void`-Funktion `sort3`, der drei Zahlen $x, y, z \in \mathbb{R}$ übergeben werden und die diese Zahlen fallend sortiert ausgibt, d.h. zuerst das Maximum $\max\{x, y, z\}$ und zuletzt das Minimum $\min\{x, y, z\}$. Schreiben Sie ferner ein aufrufendes Hauptprogramm in dem die Zahlen x, y, z eingelesen und die Funktion aufgerufen werden. Speichern Sie den Source-Code unter `sort3.c` in das Verzeichnis `serie02`.

Aufgabe 2.4. Schreiben Sie eine Funktion `folgenglied`, die für gegebenes $n \in \mathbb{N}$ das Folgenglied $a_n := (-1)^n/n$ zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem n eingelesen und a_n ausgegeben werden. Speichern Sie den Source-Code unter `folgenglied.c` in das Verzeichnis `serie02`.

Aufgabe 2.5. Schreiben Sie eine `void`-Funktion `datum`, die zu einer gegebenen Zahl $z \in \mathbb{N}$ das zugehörige Datum geeignet ausgibt. Dabei errechnet sich das gesuchte Datum aus den Ziffern von z unter Berücksichtigung der Formatierung TMMJJJJ. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem z eingelesen und die Funktion aufgerufen werden. Speichern Sie den Source-Code unter `datum.c` in das Verzeichnis `serie02`.

Aufgabe 2.6. Folgendes Programm soll zwei Ganzzahlen auf Gleichheit überprüfen. Wo liegt der Fehler?

```
#include <stdio.h>

main() {
    int x = 0, y = 0;

    printf("x = "); scanf("%d",&x);
    printf("y = "); scanf("%d",&y);

    if(x=y)
        printf("Zahlen sind gleich\n");
    else
        printf("Zahlen sind nicht gleich\n");
}
```

Aufgabe 2.7. Gegeben seien drei Punkte (x, y) , (u, v) und (a, b) in \mathbb{R}^2 . Schreiben Sie eine Funktion `punkte`, die überprüft, ob die 3 Punkte auf einer Geraden liegen. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die 6 Parameter eingelesen werden und das Resultat ausgegeben wird. Speichern Sie den Source-Code unter `punkte.c` in das Verzeichnis `serie02`.

Aufgabe 2.8. Was versteht man unter *Type-Casting*? Welche zwei Typen von *Type-Casting* gibt es? Was gibt folgender Code am Bildschirm aus? Erklären Sie auch warum!

```
#include <stdio.h>

main() {
    int x = 1;
    int y = 5;

    double erg1 = x / y;
    double erg2 = (double) x / y;
    double erg3 = 1. / 5;
    int erg4 = (double) x / y;

    printf("erg1 = %f\n",erg1);
    printf("erg2 = %f\n",erg2);
    printf("erg3 = %f\n",erg3);
    printf("erg4 = %d\n",erg4);
}
```