

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 2

Aufgabe 2.1. Schreiben Sie ein Programm, das den Radius r eines Kreises von der Tastatur einliest und die dazugehörige Fläche, sowie den Umfang berechnet und am Bildschirm ausgibt. Speichern Sie den Source-Code unter `kreis.c` in das Verzeichnis `serie02`.

Aufgabe 2.2. Ein Pythagoräisches Tripel wird von drei natürlichen Zahlen gebildet, die als Längen der Seiten eines rechtwinkligen Dreiecks vorkommen. Schreiben Sie ein Programm, das drei natürliche Zahlen $a, b, c \in \mathbb{N}$ von der Tastatur einliest und am Bildschirm ausgibt, ob es sich dabei um ein derartiges Tripel handelt oder nicht. Speichern Sie den Source-Code unter `pythagoras.c` in das Verzeichnis `serie02`.

Aufgabe 2.3. Schreiben Sie ein Programm, das drei Zahlen $x, y, z \in \mathbb{R}$ von der Tastatur einliest und diese Zahlen fallend sortiert ausgibt, d.h. zuerst das Maximum $\max\{x, y, z\}$ und zuletzt das Minimum $\min\{x, y, z\}$. Speichern Sie den Source-Code unter `sort3.c` in das Verzeichnis `serie02`.

Aufgabe 2.4. Schreiben Sie ein Programm, das die Einnahmen $E \in \mathbb{R}_0^+ := \{x \in \mathbb{R} : x \geq 0\}$, sowie die Ausgaben $A \in \mathbb{R}_0^+$ eines Unternehmens von der Tastatur einliest und am Bildschirm entweder den erzielten Gewinn bzw. Verlust ausgibt, oder dass die Einnahmen gleich den Ausgaben sind. Speichern Sie den Source-Code unter `bilanz.c` in das Verzeichnis `serie02`.

Aufgabe 2.5. Schreiben Sie ein Programm, das eine Zahl $z \in \mathbb{N}$ von der Tastatur einliest und das zugehörige Datum geeignet ausgibt. Dabei errechnet sich das gesuchte Datum aus den Ziffern von z unter Berücksichtigung der Formatierung TMMJJJJ, z.B. entspricht $z = 19102017$ dem 19. Oktober 2017. Bei unmöglichen Daten (z.B. $z = 12132016$ oder $z = 30022017$) soll das Programm eine Fehlermeldung ausgeben. Schaltjahre müssen nicht berücksichtigt werden. Speichern Sie den Source-Code unter `datum.c` in das Verzeichnis `serie02`.

Aufgabe 2.6. Schreiben Sie ein Programm, das für eine gegebene Anzahl von $n \in \mathbb{N}$ Übungen zu je 8 Beispielen die Anzahl der von Ihnen insgesamt gekreuzten Aufgaben von der Tastatur einliest. Anschließend soll am Bildschirm sowohl der von Ihnen erreichte Prozentsatz an gekreuzten Aufgaben ausgegeben werden, als auch ob sie damit eine positive Endnote in EPROG erreichen können oder nicht (vgl. <http://asc.tuwien.ac.at/eprog>). Speichern Sie den Source-Code unter `uebungsnote.c` in das Verzeichnis `serie02`.

Aufgabe 2.7. Angenommen, Sie kreuzen in EPROG im Schnitt 4 Beispiele pro Übung, dann entspricht dies 22.5 Punkten für die Berechnung der Endnote (ohne Berücksichtigung der Tafelleistungen). Kreuzen Sie im Schnitt 7 Beispiele, entspricht dies 37.5 Punkten. Allgemein berechnet sich für eine durchschnittliche Anzahl an Kreuzen $x \in [4, 8]$ die Punktezahl $y \in [22.5, 40]$ durch folgende Formel,

$$y = \min\{40, (5 + x * 10)/2\}.$$

Schreiben Sie ein Programm, das den Kreuzerschnitt $x \in [4, 8]$ von der Tastatur einliest und die dazugehörige Anzahl an Punkten y am Bildschirm ausgibt. Speichern Sie den Source-Code unter `uebungspunkte.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 = 2;
```

```
int y = 5.4;
double z = 3.6;

double erg1 = x / y + z;
double erg2 = (double) x / y;
double erg3 = 3 / 5;
int erg4 = (int) z / 2.;

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