

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 1

Aufgabe 1.1*. Legen Sie in Ihrem Home-Verzeichnis auf der `lva.student.tuwien.ac.at` ein Unterverzeichnis `serie01` an. In dieses Verzeichnis kopieren Sie bitte den Source-Code der folgenden obligatorischen Programmieraufgaben, damit Sie in der Übung nicht lange suchen müssen. Machen Sie sich in diesem Zusammenhang die Verwendung des `scp`-Befehls klar.

Aufgabe 1.2*. Schreiben Sie ein Programm `dreieck1.c`, das aus gegebenen Längen g und h (Grundseite und Höhe) den Flächeninhalt des korrespondierenden Dreiecks ausgibt.

Aufgabe 1.3*. Schreiben Sie ein Programm `skalarprodukt`, welches zu gegebenen Vektoren $\mathbf{u} = (a, b, c)$ und $\mathbf{v} = (x, y, z)$ das Skalarprodukt $w = \mathbf{u} \cdot \mathbf{v}$ mit $w = ax + by + cz$ berechnet. Die Parameter a, b, c, x, y, z sollen über die Tastatur eingelesen und das Ergebnis ausgegeben werden. Speichern Sie den Source-Code unter `skalarprodukt.c` in das Verzeichnis `serie01`.

Aufgabe 1.4*. Ändern Sie das Programm aus Aufgabe 1.2 dahingehend, dass auch das Volumen eines Prismas mit einem Dreieck als Grundfläche berechnet werden kann. Ändern Sie Ihren Code zunächst so ab, dass g und h über die Tastatur eingelesen werden können. Zusätzlich soll eine dritte Größe h_2 eingelesen werden, welche die Höhe des Prismas darstellt. Für $h_2 \neq 0$ soll das Programm nun das Volumen des Prismas ausgeben. Im Fall $h = 0$ bleibt die Ausgabe, wie zuvor, der Flächeninhalt des Dreiecks. Speichern Sie den Source-Code unter `dreieck2.c` in das Verzeichnis `serie01`.

Aufgabe 1.5*. Schreiben Sie eine Funktion `mypow`, welche aus zwei Werten a und b die Potenz a^b berechnet und zurückgibt. Gehen Sie davon aus, dass b nur die Werte 1, 2 oder 3 annehmen kann. Schreiben Sie außerdem ein aufrufendes Hauptprogramm, welches a und b von der Tastatur einliest und das Ergebnis ausgibt. Speichern Sie den Source-Code unter `mypow.c` in das Verzeichnis `serie01`.

Aufgabe 1.6. Zeichenketten (Strings) können in C durch so genannte `char`-Arrays realisiert werden. Informieren Sie sich über deren Verwendung. Schreiben Sie ein Programm, welches einen Namen (bestehend aus Vor- und Nachname) ausgibt. Der entsprechende Name soll zuvor über die Tastatur eingelesen werden. Sie können davon ausgehen, dass sowohl Vor- als auch Nachname nicht länger als 20 Zeichen sind. Speichern Sie den Source-Code unter `thatsme.c` in das Verzeichnis `serie01`.

Aufgabe 1.7. Schreiben Sie ein Programm, welches aus den 3 `int`-Werten z, n und ggT (Zähler, Nenner und grösster gemeinsamer Teiler) den Wert des Bruches $\frac{z}{n}$ berechnet und ausgibt. Achten Sie auf eventuelles Type Casting. Das Programm soll ferner mittels ggT die gekürzte Darstellung von $\frac{z}{n}$ finden und ausgeben. Speichern Sie den Source-Code unter `bruch.c` in das Verzeichnis `serie01`.

Aufgabe 1.8. Schreiben Sie eine Funktion `evalpol`, die ein Polynom dritten Grades (mit Koeffizienten a_0, a_1, a_2, a_3) an einer beliebigen Stelle x auswertet, d.h.

$$a_0 + a_1x + a_2x^2 + a_3x^3$$

berechnet. Die Koeffizienten und der Auswertungspunkt sollen aus einem aufrufendem Hauptprogramm übergeben werden. Verwenden Sie die von Ihnen geschriebene Funktion `mypow` aus Aufgabe 1.5. Speichern Sie den Source-Code unter `evalpol.c` in das Verzeichnis `serie01`.

Aufgabe 1.9. Schreiben Sie eine Funktion `myabs`, die für $x \in \mathbb{R}$ den Absolutbetrag $|x|$ zurückliefert. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem x über die Tastatur eingelesen und $|x|$ ausgegeben werden. Speichern Sie den Source-Code unter `myabs.c` in das Verzeichnis `serie01`.

Aufgabe 1.10. Schreiben Sie eine Funktion `mymax`, die von zwei Werten $x, y \in \mathbb{R}$ den größeren zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem x und y über die Tastatur eingelesen werden und das Maximum ausgegeben wird. Speichern Sie den Source-Code unter `mymax.c` in das Verzeichnis `serie01`.