

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 2

Die Aufgaben mit Stern (*) sind bis zur Übung in der kommenden Woche vorzubereiten und werden dort abgeprüft. Kopieren Sie bitte die Source-Codes auf Ihren Account auf der `lva.student.tuwien.ac.at` in ein Unterverzeichnis `serie02`. Überprüfen Sie vor der Übung, ob Ihre Source-Codes mit dem `gcc` kompiliert werden können. In den folgenden Übungsaufgaben sollen einfache **Verzweigungen** und **Zähl-schleifen** geübt werden.

Aufgabe 11*. Schreiben Sie eine 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 $\min\{x, y, z\}$. Speichern Sie den Source-Code unter `sort3.c` in das Verzeichnis `serie02`.

Aufgabe 12*. Man schreibe eine Funktion `roman`, die eine natürliche Zahl $x \in \mathbb{N}$ mit $x \leq 99$ im römischen Zahlenformat ausgibt. Zur Erinnerung:

$$C \hat{=} 100, L \hat{=} 50, X \hat{=} 10, V \hat{=} 5, I \hat{=} 1.$$

Dabei soll die Subtraktionsregel bei der Darstellung angewandt werden, d.h. schreibe `IV` statt `IIII` für 4. Man überlege sich zunächst eine Lösung für $x \leq 9$, wobei die römischen Zahlen für $1, \dots, 9$ gerade durch

$$I, II, III, IV, V, VI, VII, VIII, IX$$

gegeben sind. Auf analoge Weise schreibe man sich die 10er Ziffer auf. Ferner schreibe man ein aufrufendes Hauptprogramm, in dem die Zahl x eingelesen und die Funktion `roman` aufgerufen werden. Speichern Sie den Source-Code unter `roman.c` in das Verzeichnis `serie02`.

Aufgabe 13*. Man schreibe eine Funktion `mean`, die den Mittelwert $\frac{1}{n} \sum_{j=1}^n x_j$ eines ganzzahligen Vektors $x \in \mathbb{Z}^n$ zurückgibt. Ferner schreibe man ein aufrufendes Hauptprogramm, das den Vektor x einliest und den Mittelwert ausgibt. Die Länge n des Vektors soll eine Konstante im Hauptprogramm sein (z.B. `#DEFINE N 10`), die Funktion `mean` ist für beliebige Länge zu programmieren, d.h. n soll ein Eingangsparameter von `mean` sein. Speichern Sie den Source-Code unter `mean.c` in das Verzeichnis `serie02`.

Aufgabe 14*. Schreiben Sie eine Funktion `bin2dec`, die eine im Binärformat gegebene natürliche Zahl $0 \leq z < 256$ in eine Dezimalzahl umrechnet. Die Binärzahl werde als Vektor der Koeffizienten $a_i \in \{0, 1\}$ für $i = 0, \dots, 7$ dargestellt. Dann lässt sich der Wert der Binärzahl mittels $z = \sum_{i=0}^7 a_i 2^i$ berechnen. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Koeffizienten a_i eingelesen und der Wert von z als Dezimalzahl ausgegeben werden. Speichern Sie den Source-Code unter `bin2dec.c` in das Verzeichnis `serie02`.

Aufgabe 15. Man schreibe eine Funktion `binomial`, die mittels *einer* geeigneten Schleife den Binomialkoeffizienten $\binom{n}{k}$ berechnet. Dazu realisiere man die gekürzte Form

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

Man schreibe ein aufrufendes Hauptprogramm, in dem $k, n \in \mathbb{N}_0$ mit $k \leq n$ eingelesen werden und $\binom{n}{k}$ ausgegeben wird. Weiters schreibe man eine Lösung mit *zwei* Schleifen, bei der Zähler und Nenner getrennt berechnet werden. Welche der zwei Implementierungen ist klüger und warum?

Aufgabe 16. Gegeben seien zwei Geraden $f(x) = ax + b$ und $g(x) = cx + d$. Man schreibe eine Funktion `schnittpunkt`, die überprüft, ob sich die beiden Geraden f und g schneiden, und die ggf. den Schnittpunkt berechnet. Das Ergebnis soll geeignet ausgegeben werden. Ferner schreibe man ein Hauptprogramm, in dem die Parameter $a, b, c, d \in \mathbb{R}$ eingelesen und `schnittpunkt` aufgerufen werden.

Aufgabe 17. Gegeben seien drei Punkte (x, y) , (u, v) und (a, b) in \mathbb{R}^2 . Man schreibe eine Funktion `punkte`, die überprüft, ob die 3 Punkte auf einer Geraden liegen. Ferner schreibe man ein aufrufendes Hauptprogramm, in dem die 6 Parameter eingelesen werden und das Resultat ausgegeben wird.

Aufgabe 18. Man schreibe eine Funktion `skalarprodukt`, die zu gegebenen Vektoren $x, y \in \mathbb{R}^n$ das Skalarprodukt $x \cdot y := \sum_{j=1}^n x_j y_j$ berechnet. Ferner schreibe man ein aufrufendes Hauptprogramm, das die Vektoren x und y einliest und $x \cdot y$ ausgibt. Die Länge n der Vektoren soll eine Konstante im Hauptprogramm sein, die Funktion `skalarprodukt` ist für beliebige Länge n zu programmieren.

Aufgabe 19. Man schreibe eine Funktion `kurvendiskussion`, die für eine Parabel $p(x) = a_0 + a_1 x + a_2 x^2$ mit Koeffizientenvektor $a \in \mathbb{R}^3$ eine Kurvendiskussion durchführt. Wenn vorhanden, berechne man das Extremum (und Art) und die Nullstellen. Anderenfalls gebe man aus, dass die Parabel kein Extremum bzw. keine Nullstelle besitzt. Man schreibe ferner ein aufrufendes Hauptprogramm, das den Vektor a einliest und die Funktion aufruft.

Aufgabe 20. Man schreibe eine Funktion `dec2bin`, die zu einer natürlichen Zahl $0 \leq z < 256$ die Binärdarstellung berechnet und ausgibt. Es sollen die Koeffizienten $a_i \in \{0, 1\}$ für $i = 0, \dots, 7$ ermittelt werden, sodass $z = \sum_{i=0}^7 a_i 2^i$ gilt. Anschließend soll die Binärdarstellung in einem geeignetem Format ausgegeben werden. Beispielsweise gebe die Funktion für $z = 77$ die Zeichenfolge `0 1 0 0 1 1 0 1` aus. Ferner schreibe man ein aufrufendes Hauptprogramm, in dem z eingelesen und `dec2bin` aufgerufen werden.