Dirk Praetorius, Samuel Ferraz-Leite
Christoph Überhuber

Wintersemester 2009/10
21. Oktober 2009

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 3

Die Aufgaben mit Stern (*) sind bis zur Übung in der kommenden Woche vorzubereiten. Kopieren Sie die Source-Codes vor der Übung auf Ihren Account auf der `lva.student.tuwien.ac.at` *und überprüfen Sie, ob diese mit dem* `gcc` *compiliert werden können. In den folgenden Übungsaufgaben sollen* ***statische Arrays*** *und* ***Zählschleifen*** *geübt werden.*

Aufgabe 3.1*. Die Fibonacci-Folge ist definiert durch $x_0 := 0$, $x_1 := 1$ und $x_{n+1} := x_n + x_{n-1}$. Schreiben Sie eine Funktion `fibonacci`, die zu gegebenem Index n das Folgenglied x_n zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Index n einliest und x_n ausgibt. Speichern Sie den Source-Code im Verzeichnis `serie03` unter dem Namen `fibonacci.c`.

Aufgabe 3.2*. 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 $i = 0, \ldots, 7$ 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 im Verzeichnis `serie03` unter dem Namen `bin2dec.c`.

Aufgabe 3.3*. Gegeben sei ein Polynom $p(x) = \sum_{j=0}^{n} a_j x^j$ in Form seines Koeffizientenvektors $a = (a_0, \ldots, a_n) \in \mathbb{R}^{n+1}$. Schreiben Sie eine Funktion `evalpolynomial`, die für gegebenen Koeffizientenvektor a und Auswertungspunkt x den Funktionswert $p(x)$ berechnet. Die Funktion `pow` zur Berechnung von x^j soll *nicht* verwendet werden. Schreiben Sie eine Funktion, die möglichst nur *eine* Schleife verwendet. Der Grad $n \in \mathbb{N}$ des Polynoms soll eine Konstante im Hauptprogramm sein, die Funktion `evalpolynomial` soll aber beliebigen Grad zulassen. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Koeffizienten a_j sowie der Auswertungspunkt x eingelesen werden und $p(x)$ ausgegeben wird. Speichern Sie den Source-Code im Verzeichnis `serie03` unter dem Namen `evalpolynomial.c`.

Aufgabe 3.4*. Schreiben Sie eine Funktion `mean`, die von einem gegebenem Vektor $x \in \mathbb{N}^n$ den Mittelwert $\frac{1}{n} \sum_{j=1}^{n} x_j$ berechnet und zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor $x \in \mathbb{N}^n$ einliest und den Mittelwert ausgibt. Die Länge $n \in \mathbb{N}$ des Vektors soll eine Konstante im Hauptprogramm sein, die Funktion `mean` ist für beliebige Länge n programmieren. Speichern Sie den Source-Code im Verzeichnis `serie03` unter dem Namen `mean.c`.

Aufgabe 3.5. *Bubble-Sort* ist ein ineffizienter, aber kurzer Sortier-Algorithmus: Man vergleicht aufsteigend jedes Element eines Arrays x_j mit seinem Nachfolger x_{j+1} und - falls notwendig - vertauscht die beiden. Nach dem ersten Durchlauf muß zumindest das letzte Element bereits am richtigen Platz sein.

Der nächste Durchlauf muß also nur noch bis zur vorletzten Stelle gehen, usw. Wie viele geschachtelte Schleifen braucht dieses Vorgehen? Schreiben Sie eine Funktion `bubblesort`, die ein gegebenes Array $x \in \mathbb{R}^n$ mittels Bubble-Sort aufsteigend sortiert, d.h. $x_1 \leq x_2 \leq \cdots \leq x_n$, und zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor x einliest und in sortierter Reihenfolge ausgibt. Die Länge des Vektors soll eine Konstante im Hauptprogramm sein, die Funktion `bubblesort` ist für beliebige Länge n zu programmieren.

Aufgabe 3.6. Schriben Sie eine Funktion `energy`, die für einen gegebenen Vektor $x \in \mathbb{R}^n$ die Energie $e = \sum_{j=1}^{n} x_j^2$ zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor x einliest und die Energie ausgibt. Die Dimension $n \in \mathbb{N}$ soll eine Konstante im Hauptprogramm ein, die Funktion `energy` ist für beliebige Dimension zu programmieren.

Aufgabe 3.7. Schreiben Sie eine Funktion `maxabs`, die von einem gegebenem Vektor $x \in \mathbb{R}^n$ das erste Element x_j mit maximalem Betrag berechnet und zurückgibt, d.h. $|x_j| = \max\{|x_i| : i = 1, \ldots, n\}$ und für $|x_i| = |x_j|$ gilt $i \geq j$. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor x einliest und das Ergebnis von `maxabs` ausgibt. Die Länge des Vektors soll eine Konstante im Hauptprogramm sein, die Funktion `maxabs` ist für beliebige Länge zu implementieren.

Aufgabe 3.8. Ein Tripel $(x, y, z) \in \mathbb{N}^3$ natürlicher Zahlen heißt *pythagoräisches Zahlentripel*, falls $x^2+y^2 = z^2$ gilt. Das wohl bekannteste Beispiel ist $(3, 4, 5)$. Offensichtlich gelten $z > \max\{x, y\}$ sowie $x \neq y$ und ohne Beschränkung der Allgemeinheit ferner $x < y$. Schreiben Sie eine `void`-Funktion `pythagoras`, die zu gegebener Schranke $n \in \mathbb{N}$ alle pythagoräischen Zahlentripel mit $x < y < z \leq n$ bestimmt und ausgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Schranke n eingelesen und `pythagoras` aufgerufen wird.

Aufgabe 3.9. Für $p \in [1, \infty)$ ist die ℓ_p-Norm auf $\mathbb{R}^n$ definiert durch

$$\|x\|_p := \Big(\sum_{j=1}^{n} |x_j|^p \Big)^{1/p}.$$

Schreiben Sie eine Funktion `pnorm`, die einen Vektor $x \in \mathbb{R}^n$, dessen Länge n sowie $p \in [1, \infty)$ übernimmt und $\|x\|_p$ zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem x und p eingelesen werden und $\|x\|_p$ ausgegeben wird. Die Dimension $n \in \mathbb{N}$ soll eine Konstante im Hauptprogramm sein, die Funktion `pnorm` soll aber beliebige Dimension zulassen. Testen Sie Ihr Programm mit verschiedenen Werten für p bei festem Vektor x. Was beobachten Sie für $p \to \infty$?

Aufgabe 3.10. Schreiben Sie eine `void`-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 $i = 0, \ldots, 7$ 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. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem z eingelesen und `dec2bin` aufgerufen werden.