Dirk Praetorius, Samuel Ferraz-Leite
Christoph Überhuber Sommersemester 2010
13. April 2010

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 4

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 **dynamische Arrays** und **Schleifen** geübt werden.*

Aufgabe 4.1*. Schreiben Sie eine Funktion `prim`, die überprüft, ob eine natürliche Zahl $n \in \mathbb{N}$ eine Primzahl ist (Rückgabewert 1) oder nicht (Rückgabewert 0). Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Wert n einliest und ausgibt, ob es sich um eine Primzahl handelt. Speichern Sie den Source-Code im Verzeichnis `serie04` unter dem Namen `prim.c`.

Aufgabe 4.2*. Die Gleitpunkt-Arithmetik ist nicht assoziativ. Das numerische Ergebnis der Summe $\sum_{j=0}^{n} x^j/j!$ hängt daher von der Summationsreihenfolge ab. Schreiben Sie eine Funktionen `forward` und `backward`, die diese Summe auf zwei Weisen berechnen: `forward` entspricht Vorwärtssummation $j = 0, \dots, n$, `backward` entspricht Rückwärtssummation $j = n, \dots, 0$. Welche Variante wird besser sein – zumindest für positives x und großes n? Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Schranke $n \in \mathbb{N}$ eingelesen und die Ergebnisse von `forward` bzw. `backward` ausgegeben werden. Speichern Sie den Source-Code im Verzeichnis `seie04` unter dem Namen `forwardbackward.c`.

Aufgabe 4.3*. Die Frobeniusnorm einer Matrix $A \in \mathbb{R}^{m \times n}$ ist durch

$$\|A\|_F := \Big(\sum_{j=1}^{m} \sum_{k=1}^{n} A_{jk}^2 \Big)^{1/2}$$

definiert. Schreiben Sie eine Funktion `frobeniusnorm`, die für gegebene Matrix A und gegebene Dimensionen $m, n \in \mathbb{N}$ die Frobeniusnorm berechnet. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Zeilen- und Spaltendimensionen $m, n \in \mathbb{N}$ und A eingelesen werden und $\|A\|_F$ ausgegeben wird. Die Matrix A soll dabei als dynamische Matrix (vom Typ `double**`) realisiert werden. Speichern Sie den Source-Code im Verzeichnis `serie04` unter dem Namen `frobeniusnorm.c`.

Aufgabe 4.4*. Gegeben seien Polynome $p(x) = \sum_{j=0}^{m} a_j x^j$ und $q(x) = \sum_{j=0}^{n} b_j x^j$ in Form ihrer Koeffizientenvektoren $a \in \mathbb{R}^{m+1}$ und $b \in \mathbb{R}^{n+1}$. Dann ist die Summe $r = p + q$ ein Polynom vom Grad $\max\{m, n\}$. Schreiben Sie eine Funktion `addpolynomials`, die für gegebene Koeffizientenvektoren a und b den Koeffizientenvektor c von r anlegt, berechnet und zurückgibt. Im aufrufenden Hauptprogramm sollen $m, n \in \mathbb{N}$ sowie die dynamischen Vektoren $a \in \mathbb{R}^m$ und $b \in \mathbb{R}^n$ eingelesen werden und c ausgegeben werden.

Aufgabe 4.5. Schreiben Sie eine Funktion `transpose`, die zu einer dynamisch gespeicherten Matrix $A \in \mathbb{R}^{m \times n}$ (vom Typ `double**`) die transponierte Matrix $A^T \in \mathbb{R}^{n \times m}$ berechnet. Dabei sind die Einträge von A^T gerade durch $(A^T)_{jk} = A_{kj}$ definiert. Im Hauptprogramm sollen die Dimensionen $m, n \in \mathbb{N}$ sowie die Einträge der Matrix A eingelesen und A^T ausgegeben werden.

Aufgabe 4.6. In vielen mathematischen Bibliotheken werden Matrizen $A \in \mathbb{R}^{m\times n}$ spaltenweise gespeichert, d.h. in Form eines Vektors $a \in \mathbb{R}^{mn}$, wobei $a_{j+km} = A_{jk}$ gilt, wenn die Indizierung (wie in C üblich) bei 0 beginnt. Schreiben Sie eine Funktion `mvmultiplication`, die die Matrix-Vektor-Multiplikation einer spaltenweise gespeicherten Matrix $A \in \mathbb{R}^{m\times n}$ mit einem Vektor $x \in \mathbb{R}^n$ realisiert. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem A und x eingelesen werden und $b = Ax$ ausgegeben wird.

Aufgabe 4.7. Programmieren Sie die Funktion `transpose` aus Aufgabe 4.5 erneut. Diesmal soll die Funktion jedoch mit spaltenweise gespeicherten Matrizen (vom Typ `double*`) wie in Aufgabe 4.6 beschrieben arbeiten. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Dimensionen $m, n \in \mathbb{N}$ sowie die Einträge A_{jk} eingelesen und die transponierte Matrix A^T ausgegeben werden.

Aufgabe 4.8. 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 Grad n, 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. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem der Grad $n \in \mathbb{N}$ und die Koeffizienten a_j sowie der Auswertungspunkt x eingelesen werden und $p(x)$ ausgegeben wird.

Aufgabe 4.9. Eine Matrix $A \in \mathbb{R}^{n\times n}$ ist symmetrisch, falls $A_{jk} = A_{kj}$ für alle $j, k = 1, \ldots, n$ gilt. Schreiben Sie eine Funktion `issymmetric`, die eine Matrix A auf Symmetrie überprüft (Rückgabewert `1` bei Symmetrie und `0` bei Nicht-Symmetrie). Schreiben Sie ein aufrufendes Hauptprogramm, in dem A eingelesen wird und ausgegeben wird, ob A symmetrisch ist oder nicht. Speichern Sie A als C-Matrix, also vom Typ `double**`.

Aufgabe 4.10. Für eine Matrix $A \in \mathbb{R}^{m\times n}$ ist die Zeilensummennorm durch

$$\|A\| = \max_{j=1,\ldots,m} \sum_{k=1}^{n} |A_{jk}|$$

gegeben. Schreiben Sie eine Funktion `zeilensummennorm`, die die Zeilensummennorm einer Matrix A berechnet. Speichern Sie die Matrix spaltenweise, vgl. Aufgabe 4.6. Schreiben Sie ein aufrufendes Hauptprogramm, in dem die Dimensionen m, n sowie die Einträge der Matrix A eingelesen und $\|A\|$ ausgegeben werden.