

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 4

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Die übrigen Aufgaben dienen nur Ihrer Übung und mir als zusätzliche Grundlage für den Prüfungsstoff in den schriftlichen Tests. Kopieren Sie bitte den Source-Code in ein Unterverzeichnis `serie04` Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit dem `gcc` kompiliert werden können. — Vergessen Sie beim Kompilieren ggf. bitte nicht die Option `-lm`, um die mathematische Bibliothek zu linken. — In den folgenden Aufgaben sollen **Zähl- und Bedingungsschleifen** geübt werden.

Aufgabe 31*. Für $p \in [1, \infty)$ ist die ℓ_p -Norm auf \mathbb{R}^n definiert durch

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

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 soll eine Konstante im Hauptprogramm sein. Speichern Sie den Source-Code unter `pnorm.c` ins Verzeichnis `serie04`.

Aufgabe 32*. 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. Für eine Matrix $A \in \mathbb{R}^{m \times n}$ ist die Zeilensummennorm durch

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

gegeben. Schreiben Sie eine Funktion `zeilensummennorm`, die die Zeilensummennorm einer Matrix A berechnet, die spaltenweise gespeichert ist. Schreiben Sie ein aufrufendes Hauptprogramm, in dem A eingelesen und $\|A\|$ ausgegeben wird. Die Dimensionen $m, n \in \mathbb{N}$ der Matrix sollen Konstanten im Hauptprogramm, aber Parameter der Funktion `zeilensummennorm` sein. Speichern Sie den Source-Code unter `zeilensummennorm.c` ins Verzeichnis `serie04`.

Aufgabe 33*. Man schreibe eine Funktion `wurzelschranke`, die zu einer gegebenen Zahl $x \geq 0$ die natürliche Zahl $k \in \mathbb{N}_0$ mit $k \leq \sqrt{x} < k + 1$ zurückgibt. Dabei dürfen weder die Wurzel-Funktion `sqrt`, noch Rundungsoperationen (z.B. `floor` oder `ceil` etc.) verwendet werden. Speichern Sie den Source-Code unter `wurzelschranke.c` ins Verzeichnis `serie04`.

Aufgabe 34*. Die Sinus-Funktion hat die Reihendarstellung

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

Wir betrachten die Partialsummen

$$S_n(x) = \sum_{k=0}^n (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

Man schreibe eine C-Funktion `sin_`, die für gegebene $x \in \mathbb{R}$ und $\varepsilon > 0$ den Wert $S_n(x)$ zurückliefert, sobald

$$|S_n(x) - S_{n-1}(x)|/|S_n(x)| \leq \varepsilon \quad \text{oder} \quad |S_n(x)| \leq \varepsilon$$

gilt. Man schreibe ferner ein aufrufendes Hauptprogramm, in dem $x \in \mathbb{R}$ und $\varepsilon > 0$ eingelesen werden. Neben dem berechneten Wert $S_n(x)$ sollen auch der korrekte Wert $\sin(x)$ und der absolute Fehler $|S_n(x) - \sin(x)|$ ausgegeben werden sowie der relative Fehler $|S_n(x) - \sin(x)|/|\sin(x)|$ im Fall $\sin(x) \neq 0$. Speichern Sie den Source-Code unter `sin.c` ins Verzeichnis `serie04`.

Aufgabe 35. Man schreibe die Funktion in Aufgabe 34 so, dass diese mit einer Schleife auskommt und dass x^{2k+1} und $(2k+1)!$ möglichst kostensparend realisiert werden. Man verwende also insbesondere keine (vor- oder selbst implementierten) Funktionen zur Berechnung der Potenz oder der Faktoriellen.

Aufgabe 36. Schreiben Sie eine Funktion `mvmultiplication`, die die Matrix-Vektor-Multiplikation einer spaltenweise gespeicherten Matrix $A \in \mathbb{R}^{m \times n}$ realisiert mit einem Vektor $x \in \mathbb{R}^n$ realisiert und das Ergebnis in einen Vektor $b \in \mathbb{R}^m$ schreibt, der mittels Call by Reference übergeben wird. Die Dimensionen $m, n \in \mathbb{N}$ sollen Konstanten im Hauptprogramm sein, aber als Parameter an die Funktion übergeben werden. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem A und x eingelesen werden und $b = Ax$ ausgegeben wird.

Aufgabe 37. Schreiben Sie eine Funktion `transpose`, die zur einer spaltenweise gespeicherten Matrix $A \in \mathbb{R}^{m \times n}$ die transponierte $A^T \in \mathbb{R}^{n \times m}$ berechnet. Dabei soll die Matrix A^T ebenfalls spaltenweise gespeichert werden, und die Einträge der Matrix A sollen mit den Einträgen der Matrix A^T überschrieben werden. Die Dimensionen $m, n \in \mathbb{N}$ sollen Konstanten im Hauptprogramm sein, aber als Parameter an die Funktion übergeben werden. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem A eingelesen und A^T ausgegeben wird.

Aufgabe 38. Man schreibe eine verbesserte Variante von Bubble-Sort, bei dem man sich die Stelle des letzten Tausches merkt. Ab dieser Stelle müssen die Daten ja bereits sortiert sein. Der nächste Durchlauf braucht also nur noch bis zu dieser Array-Position zu gehen. Zur Beschreibung von Bubble-Sort siehe Aufgabe 28.

Aufgabe 39. Was ist ein Gleichkommazahlensystem? Aus welchen Bestandteilen setzt sich eine Gleitkommazahl zusammen? Wie bestimmt man daraus ihren Wert? Was verbirgt sich hinter den Symbolen `Inf`, `-Inf` und `NaN`? Was ist die Maschinengenauigkeit `eps`? Was ist eine denormalisierte Gleitkommazahl? Was ist ein implizites erstes Bit?

Aufgabe 40. Wieviele Ganzzahlen kann man mit n Bytes darstellen. Welcher Zahlbereich wird in der Regel abgedeckt? Was passiert, wenn man durch arithmetische Operationen diesen Bereich verlässt?