

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 5

Aufgabe 5.1*. Schreiben Sie eine Funktion `double powN(double x, int n)`, welche x^n für einen ganzzahligen Exponenten $n \in \mathbb{Z}$ berechnet. Es gilt $x^0 = 1$ für alle $x \in \mathbb{R}$ und für $n < 0$ gilt $x^n = (1/x)^{-n}$. Weiters gilt $0^n = 0$ für $n > 0$. Die Potenz 0^n ist für $n < 0$ nicht definiert. Die Funktion soll in diesem Fall den Wert `0.0/0.0` zurückgeben. Speichern Sie den Source-Code unter `powN.c` in das Verzeichnis `serie05`.

Aufgabe 5.2*. Schreiben Sie eine `void`-Funktion `diffpol` welche die Koeffizienten der Ableitung eines gegebenen Polynoms

$$p(x) = \sum_{j=0}^n a_j x^j \quad \text{mit} \quad p'(x) = \sum_{j=0}^{n-1} b_j x^j$$

berechnet. Welche Eingabeparameter benötigen Sie hierzu? Der Koeffizientenvektor soll schließlich ausgegeben werden. Testen sie ihren Code in einem Hauptprogramm. Dabei soll die Länge des Polynoms p eine Konstante in ihrem Hauptprogramm sein. Die Funktion `diffpol` soll aber für beliebiges n geschrieben werden. Speichern Sie den Source-Code unter `diffpol.c` in das Verzeichnis `serie05`.

Aufgabe 5.3*. Die Frobeniusnorm einer Matrix $A \in \mathbb{R}^{M \times N}$ ist durch

$$\|A\|_F^2 := \sum_{j=1}^M \sum_{k=1}^N A_{jk}^2$$

definiert. Schreiben Sie eine Funktion `frobeniusnorm`, die für gegebene Matrix A und gegebene Größen $M, N \in \mathbb{N}$ die Frobeniusnorm berechnet und zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem A eingelesen wird und $\|A\|_F^2$ ausgegeben wird. Die Größen M, N sollen dabei feste Konstanten im Hauptprogramm sein, die Funktion `frobeniusnorm` soll aber für beliebiges m, n programmiert werden. Die Matrix A soll dabei spaltenweise in einem Array gespeichert werden. Speichern Sie den Source-Code unter `frobeniusnorm.c` in das Verzeichnis `serie05`.

Aufgabe 5.4*. Die Sinus-Funktion hat die Reihendarstellung

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

Wir betrachten die Partialsummen

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

Schreiben Sie eine 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. Schreiben Sie 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` in das Verzeichnis `serie05`.

Aufgabe 5.5. Schreiben Sie eine `void`-Funktion `vielfache(k, nmax)`, die alle ganzzahligen Vielfachen der Zahl $k \in \mathbb{N}$ mit $k \leq n_{\max} \in \mathbb{N}$ am Bildschirm ausgibt. Die Ausgabe erfolge zeilenweise in der Form

```
1 x 5 = 5
2 x 5 = 10
3 x 5 = 15
```

beispielsweise für den Fall $k = 5$ und $n_{\max} = 19$. Ferner schreibe man ein Hauptprogramm, das die Daten k und n von der Tastatur einliest und `vielfache(k, nmax)` aufruft. Speichern Sie den Source-Code unter `vielfache.c` in das Verzeichnis `serie05`.

Aufgabe 5.6. Schreiben Sie eine `void`-Funktion `pascal(k)`, die die ersten k -Stufen des Pascal'schen Dreiecks ausgibt: Jede Zeile dieses Schemas beginnt und endet mit 1. Die restlichen Zahlen werden als Summe nebeneinanderstehender Zahlen der vorhergegangenen Zeile gebildet. Für $k = 5$ gilt beispielsweise

```

          1
         1 1
        1 2 1
       1 3 3 1
      1 4 6 4 1
```

Schreiben Sie ferner ein Hauptprogramm, das $k \in \mathbb{N}$ einliest und die Funktion aufruft. Speichern Sie den Source-Code unter `pascal.m` in das Verzeichnis `serie05`.

Aufgabe 5.7. Schreiben Sie ein Programm, welche mehrere *Lotto 6 aus 45* Tipps einliest und überprüft ob der richtige Tipp dabei war. Das Programm soll k Tipps einlesen und in einer $k \times 6$ Matrix speichern. Die Anzahl der Tipps soll dabei eine feste Konstante im Hauptprogramm sein. Danach soll noch ein Vektor a mit dem "richtigen" Tipp eingelesen werden. Überprüfen Sie danach ob Sie einen Lotto 6er haben! *Optional* können Sie Ihr Programm noch erweitern und überprüfen ob einer Ihrer Tipps 3, 4 oder 5 richtige Zahlen enthält. Speichern Sie den Source-Code unter `tipps.c` in das Verzeichnis `serie05`.

Hinweis: Teilen Sie ihr Programm in mehrere Funktionen auf. Beispielsweise soll eine Funktion die Tipps einlesen, eine andere Funktion soll überprüfen ob ein richtiger Tipp dabei war. Weiters ist es vermutlich sinnvoll die eingelesenen Tipps aufsteigend zu sortieren. Beachten Sie ausserdem, dass in einem Tipp eine Zahl maximal einmal vorkommen darf. Das Programm soll das natürlich berücksichtigen und überprüfen (gleich beim Einlesen)!

Aufgabe 5.8. Schreiben Sie eine Funktion `power`, die für gegebene reelle Zahlen $x > 1$ und $C > 0$ die kleinste Zahl $n \in \mathbb{N}$ berechnet mit $x^n > C$. Dabei soll die Funktion `log` nicht verwendet werden. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem x und C eingelesen werden und n ausgegeben wird. Speichern Sie den Source-Code unter `power.c` in das Verzeichnis `serie05`.