

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 5

Aufgabe 5.1*. Schreiben Sie eine Funktion `float2dec`, die für eine gegebene Mantissenlänge $M \in \mathbb{N}$, Ziffern $a_1, \dots, a_M \in \{0, 1\}$ und einen Exponenten $e \in \mathbb{Z}$ den Dezimalwert $x = (\sum_{k=1}^M a_k 2^{-k}) 2^e$ berechnet und zurückgibt. Schreiben Sie ein aufrufendes Hauptprogramm, in dem M , a_j und e eingelesen und der Wert x ausgegeben wird. Realisieren die Potenzen 2^{-k} möglichst rechenökonomisch. Speichern Sie den Source-Code unter `float2dec.c` in das Verzeichnis `serie05`.

Aufgabe 5.2*. 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 und den Ergebnisvektor $b = Ax \in \mathbb{R}^m$ zurückgibt. Die Dimensionen $m, n \in \mathbb{N}$ können Konstanten im Hauptprogramm sein, müssen 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. Speichern Sie den Source-Code unter `mvmultiplication.c` in das Verzeichnis `serie05`.

Aufgabe 5.3*. Schreiben Sie einen Strukturdatentyp `cdouble`, in dem Realteil und Imaginärteil einer Zahl $a + bi \in \mathbb{C}$ jeweils als `double` gespeichert werden. Schreiben Sie Funktionen `newCDouble`, `delCDouble` sowie die vier Zugriffsfunktionen `setCDoubleReal`, `getCDoubleReal`, `setCDoubleImag` sowie `getCDoubleImag`. Speichern Sie den Source-Code, aufgeteilt in Header-Datei `cdouble.h` und `cdouble.c`, ins Verzeichnis `serie05`.

Aufgabe 5.4*. Schreiben Sie Funktionen, die die Addition, die Subtraktion, die Multiplikation und die Division für komplexe Zahlen $a + bi \in \mathbb{C}$ realisieren. Verwenden Sie zur Speicherung die Struktur aus Aufgabe 5.3, und benutzen Sie beim Strukturzugriff nur die entsprechenden Zugriffsfunktionen. Schreiben Sie ein aufrufendes Hauptprogramm, in dem zwei komplexe Zahlen $w, z \in \mathbb{C}$ eingelesen werden und $w + z$, $w - z$, $w \cdot z$ sowie w/z ausgegeben werden. Binden Sie den Code aus Aufgabe 5.3 mittels `#include "cdouble.c"` ein. Speichern Sie den Source-Code unter `carithmetik.c` in das Verzeichnis `serie05`.

Aufgabe 5.5. Schreiben Sie einen Strukturdatentyp `polynomial` zur Speicherung von Polynomen, die bezüglich der Monombasis dargestellt sind, d.h. $p(x) = \sum_{j=0}^n a_j x^j$. Es ist also der Grad $n \in \mathbb{N}_0$ sowie der Koeffizientenvektor $(a_0, \dots, a_n) \in \mathbb{R}^{n+1}$ zu speichern. Schreiben Sie alle nötigen Funktionen, um mit dieser Struktur arbeiten zu können (`newPoly`, `delPoly`, `getPolyDegree`, `getPolyCoefficient`, `setPolyCoefficient`).

Aufgabe 5.6. Das Produkt $r = pq$ zweier Polynome $p(x) = \sum_{j=0}^m a_j x^j$ und $q(x) = \sum_{j=0}^n b_j x^j$ ist wieder ein Polynom. Schreiben Sie eine Funktion `prodPoly`, die das Produktpolynom r berechnet und in der Struktur aus Aufgabe 5.5 speichert. Überlegen Sie sich zunächst, welchen Grad das Polynom r hat

und wie sich die Koeffizienten berechnen lassen. Schreiben Sie ein aufrufendes Hauptprogramm, in dem p und q eingelesen und $r = pq$ ausgegeben wird.

Aufgabe 5.7. Schreiben Sie eine Struktur `CPoly` zur Speicherung von Polynomen mit komplexwertigen Koeffizienten, die bezüglich der Monombasis dargestellt sind, d.h. $p(x) = \sum_{j=0}^n a_j x^j$. Es sind also der Grad $n \in \mathbb{N}_0$ sowie der Koeffizientenvektor $(a_0, \dots, a_n) \in \mathbb{C}^{n+1}$ zu speichern. Verwenden Sie für die Darstellung der komplexwertigen Koeffizienten den Strukturdatentyp aus Aufgabe 5.3. Schreiben Sie die ferner die nötigen Zugriffsfunktionen `newCPoly`, `delCPoly`, `getCPolyDegree`, `getCPolyCoefficient` und `setCPolyCoefficient`.

Aufgabe 5.8. Modifizieren Sie die Funktion aus Aufgabe 5.6, sodass das Produkt für Polynome mit komplexen Koeffizienten berechnet wird. Verwenden Sie hierzu die Struktur und die Funktionen aus den Aufgaben 5.7 und 5.4.

Aufgabe 5.9. Schreiben Sie eine Funktion `dec2float`, die für eine gegebene Dezimalzahl $x \in \mathbb{R}_{>0}$ und eine Mantissenlänge $M \in \mathbb{N}$ die Ziffern $a_1, \dots, a_M \in \{0, 1\}$ und den Exponenten $e \in \mathbb{Z}$ der normalisierten Gleitkommadarstellung (d.h. $a_1 = 1$) berechnet und zurückgibt. Schreiben Sie ein aufrufendes Hauptprogramm, in dem x eingelesen und die Gleitkommadarstellung von x ausgegeben wird. Um Ihre Funktion zu verifizieren, können Sie Aufgabe 5.1 verwenden.

Aufgabe 5.10. Schreiben Sie eine Funktion `cmatrixvector`, die die Matrix-Vektor-Multiplication $Ax \in \mathbb{C}^m$ mit einer Matrix $A \in \mathbb{C}^{m \times n}$ und einem Vektor $x \in \mathbb{C}^n$ realisiert. Verwenden Sie für die Koeffizienten die komplexe Arithmetik aus Aufgabe 5.3.