

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 6

Aufgabe 6.1*. 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. Die Mantissenlänge M soll hierbei eine Konstante im Hauptprogramm sein. Schreiben Sie ein aufrufendes Hauptprogramm, in dem 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 `serie06`.

Aufgabe 6.2*. 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 am Bildschirm ausgibt. Schreiben Sie ein aufrufendes Hauptprogramm, in dem x eingelesen wird. Um Ihre Funktion zu verifizieren, können Sie Aufgabe 6.1 verwenden. Speichern Sie den Source-Code unter `dec2float.c` in das Verzeichnis `serie06`.

Hinweis: Durch die feste Mantissenlänge $M \in \mathbb{N}$ kann x evtl. nur näherungsweise wiedergegeben werden.

Aufgabe 6.3*. Was ist der Unterschied und der Zusammenhang zwischen einer Variable und einem Pointer? Was könnten Vor- und Nachteile dieser Konstrukte sein? Schreiben Sie eine Funktion `swap`, welche die Werte zweier Zahlen a und b vertauscht. Warum funktioniert das folgende Vorgehen nicht?

```
void swap(double x, double y)
{
    double tmp;
    tmp = x;
    x = y;
    y = tmp;
}
```

Speichern Sie den Source-Code unter `swap` in das Verzeichnis `serie06`.

Aufgabe 6.4*. Erweitern Sie die Funktion `minmax` aus der Vorlesung (Folie 105) folgendermaßen: Neben dem Minimum und Maximum des Vektors, sollen auch noch der *erste* Index j bei dem das Minimum angenommen wird und der *erste* Index k bei dem das Maximum angenommen wird, mittels *Call by Reference* zurückgegeben werden. Speichern Sie den Source-Code unter `minmax2.c` in das Verzeichnis `serie06`.

Aufgabe 6.5. Für $x > 0$ konvergiert die Folge

$$x_1 := \frac{1}{2}(1+x), \quad x_{n+1} := \frac{1}{2}\left(x_n + \frac{x}{x_n}\right) \quad \text{für } n \geq 1$$

gegen \sqrt{x} . Schreiben Sie eine Funktion `sqrt_`, die für gegebene $x > 0$ und $\tau > 0$ als Ergebnis das erste Folgenglied $y = x_n$ zurückgibt, für das gilt

$$\frac{|x_n - x_{n+1}|}{|x_n|} \leq \tau \quad \text{oder} \quad |x_n| \leq \tau.$$

Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem x eingelesen und neben der Approximation x_n von \sqrt{x} auch der exakte Wert sowie der absolute Fehler $|x_n - \sqrt{x}|$ ausgegeben werden. Speichern Sie den Source-Code unter `sqrt.c` in das Verzeichnis `serie06`.

Aufgabe 6.6. Gegeben Sei ein sortierter `int` Vektor x der Länge n . Schreiben Sie eine Funktion `find(x,y)`, die einen Index i zurückgibt, für den $x(i) = y$ gilt. Falls y nicht in x vorkommt, werde -1 zurückgegeben. Speichern Sie den Source-Code unter `find.c` in das Verzeichnis `serie06`.

Aufgabe 6.7. In der Vorlesung haben Sie gelernt, dass es zu jeder Zahl $x \in \mathbb{R}$

- ein Vorzeichen $\sigma \in \{\pm 1\}$,
- Ziffern $a_j \in \{0, 1\}$ und
- einen Exponenten $e \in \mathbb{Z}$

gibt, so dass

$$x = \sigma \left(\sum_{k=1}^{\infty} a_k 2^{-k} \right) 2^e$$

gilt (Folie 95). Machen Sie sich den Beweis dieser Aussage klar und geben Sie ihn in eigenen Worten wieder. Diese Darstellung ist im Allgemeinen nicht eindeutig. Sie kann aber in einem Gleitkommazahlensystem $\mathbb{F}(2, M, e_{\min}, e_{\max})$ durch feste Mantissenlänge $M \in \mathbb{N}$ und Normierung $a_1 = 1$ eindeutig gemacht werden. Warum? Beweisen Sie diese Aussage. Was wissen Sie über das implizite erste Bit?

Aufgabe 6.8. Was ist die größte, was die kleinste Zahl, die von einer Variable des Typs `long` gespeichert werden kann? Schreiben Sie ein Programm, welches Die Bit-Größe von `long` berechnet und die beiden Zahlen ausgibt. Erzeugen Sie die gleiche Ausgabe schließlich für den Datentyp `unsigned int`. Speichern Sie den Source-Code unter `big.c` in das Verzeichnis `serie06`.