Dirk Praetorius,
Thomas Führer

Sommersemester 2014
24.03.2014

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 4

Aufgabe 4.1. Schreiben Sie eine nicht-rekursive Funktion `binomial`, die den Binomialkoeffizienten $\binom{n}{k}$ berechnet. Dazu realisiere man die gekürzte Form $\binom{n}{k} = \frac{n\cdot(n-1)\cdots(n-k+1)}{1\cdot 2\cdots k} = \frac{n}{1}\cdot\frac{n-1}{2}\cdots\frac{n-k+1}{k}$ mittels geeigneter Schleifen. Schreiben Sie ein aufrufendes Hauptprogramm, in dem $k, n \in \mathbb{N}_0$ mit $k \le n$ eingelesen werden und $\binom{n}{k}$ ausgeben wird. Speichern Sie den Source-Code unter `binomial.c` in das Verzeichnis `serie04`.

Aufgabe 4.2. Lösen Sie diese Aufgabe mit einer rekursiven Funktion! Für $x > 0$ konvergiert die Folge

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

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

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

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 `serie04`.
Hinweis: Zur Berechnung von $\sqrt{x}$ können Sie die Funktion `sqrt` aus der Mathematikbibliothek verwenden. Um den Absolutbetrag einer reellen Zahl zu bestimmen, darf die Funktion `fabs` aus der Mathematikbibliothek verwendet werden.

Aufgabe 4.3. Sie legen ihr Kapital bei ihrer Hausbank zu einem fixen Jahreszinssatz an. Schreiben Sie eine Funktion `endkapital` welches aus den Werten Laufzeit $n \in \mathbb{N}$, Jahreszinssatz p (in Prozent %), und Startkapital $x \in \mathbb{R}_{\ge 0}$ das Endkapital nach n Jahren zurückgibt. Dabei soll die Funktion ihren Kontostand in folgender Form

```
Jahr    Kapital
====    =======
0      1000.00
1      1010.00
2      1020.10
3      1030.30
..      .......
10      1104.62
```

ausgeben. (Bei diesem Beispiel wurde $p = 1$, $n = 10$, und $x = 1000.00$ gewählt.) Schreiben Sie ferner eine Funktion `laufzeit`, welche berechnet wie lange Sie mindestens Ihr Startkapital x bei einem Zinssatz p anlegen müssen um ein Endkapital von mindestens $x_{\max}$ zu haben. Die Funktion soll x, p, und $x_{\max}$ als Eingabe erhalten. Weiters schreiben Sie ein Hauptprogramm welches die beiden Funktionen testet. Wie lange müssen Sie warten um Euromillionär zu werden, wenn Sie $x = 1000$ Euro bei einem Fixzinssatz von $p = 4$ anlegen? Speichern Sie den Source-Code unter `kapital.c` in das Verzeichnis `serie04`.

Aufgabe 4.4. 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. Für diese Aufgabe dürfen Sie die Funktion `pow` aus der Mathematikbibliothek nicht verwenden. Speichern Sie den Source-Code unter `powN.c` in das Verzeichnis `serie04`.

Aufgabe 4.5. Um die Summe $\sum_{j=1}^{n}(-1)^j/j$ zu berechnen, ist es numerisch günstig, zunächst die negativen und die positiven Beiträge getrennt zu summieren und erst abschließend beide Teilsummen zu addieren. Warum? Schreiben Sie eine Funktion `sum`, die dieses Vorgehen realisiert. Schreiben Sie ferner ein Hauptprogramm, das n einliest und $\sum_{j=1}^{n}(-1)^j/j$ ausgibt. Speichern Sie den Source-Code unter `sum.c` in das Verzeichnis `serie04`.

Aufgabe 4.6. Schreiben Sie eine Funktion `maxabs`, die von einem gegebenem Vektor $x \in \mathbb{R}^n$ das erste Element x_j mit maximalem Betrag berechnet und zurückgibt, d.h. $|x_j| = \max\{|x_i| \,:\, i = 1, \dots, n\}$ und falls $|x_i| = |x_j|$ dann gilt $i \geq j$. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor x einliest und das Ergebnis von `maxabs` ausgibt. Der Vektor x soll dabei mittels statischem Array realisiert werden. Die Länge des Vektors soll eine Konstante im Hauptprogramm sein, die Funktion `maxabs` ist aber für beliebige Länge zu implementieren! Speichern Sie den Source-Code unter `maxabs.c` in das Verzeichnis `serie04`.

Aufgabe 4.7. Schreiben Sie eine Funktion `geometricMean`, die von einem gegebenem Vektor $x \in \mathbb{R}^n_{\geq 0}$ den geometrischen Mittelwert

$$\overline{x}_{\text{geom}} = \sqrt[n]{\prod_{j=1}^{n} x_j}$$

berechnet und zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor $x \in \mathbb{R}^n$ einliest und den geometrischen Mittelwert ausgibt. Die Länge $n \in \mathbb{N}$ des Vektors soll eine Konstante im Hauptprogramm sein, die Funktion `geometricMean` ist für beliebige Länge n zu programmieren. Speichern Sie den Source-Code unter `geometricMean.c` in das Verzeichnis `serie04`.

Aufgabe 4.8. Folgendes Programm soll den maximalen Eintrag einer gegebenen Matrix bestimmen. Als Ergebnis wird `5.0000` ausgegeben. Wo liegt der Fehler?

```
#include <stdio.h>

main() {
  double A[2][3] = { {1,2,3},{6,-4,5} };
  double max = A[0][0];

  int j=0, k=0;

  for(j=0; j<2; j=j+1) {
    for(k=1; k<3; k=k+1) {
      if(A[j][k] > max) {
        max = A[j][k];
      }
    }
  }
  printf("Maximum = %f\n",max);
}
```

Beheben Sie den Fehler und erweitern Sie das Programm so, dass auch das Minimum aller Matrixeinträge bestimmt wird. Speichern Sie den Source-Code unter `maxminmatrix.c` in das Verzeichnis `serie04`.