

## Übungen zur Vorlesung Einführung in das Programmieren für TM

### Serie 7

**Aufgabe 7.1.** Die Quotientenfolge  $(a_{n+1}/a_n)_{n \in \mathbb{N}}$  zur Fibonacci-Folge  $(a_n)_{n \in \mathbb{N}}$ ,

$$a_0 := 1, \quad a_1 := 1, \quad a_n := a_{n-1} + a_{n-2} \quad \text{für } n \geq 2,$$

konvergiert gegen den goldenen Schnitt  $(1 + \sqrt{5})/2$ . Insbesondere konvergiert die Differenz

$$b_n := \frac{a_{n+1}}{a_n} - \frac{a_n}{a_{n-1}}$$

gegen Null. Schreiben Sie eine Funktion `cauchy`, die zu gegebenem  $k \in \mathbb{N}$  die kleinste Zahl  $n \in \mathbb{N}$  mit  $|b_n| \leq 1/k$  zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das die Zahl  $k \in \mathbb{N}$  einliest und den zugehörigen Index  $n \in \mathbb{N}$  ausgibt. Speichern Sie den Source-Code unter `goldenerSchnitt.c` in das Verzeichnis `serie07`.

**Aufgabe 7.2.** Die Cosinus-Funktion hat die Darstellung  $\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!}$ . Wir betrachten die Partialsummen

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

Schreiben Sie eine Funktion `cos_`, die für gegebene  $x \in \mathbb{R}$  und  $\tau > 0$  den Wert  $C_n(x)$  zurückliefert, sobald

$$|C_n(x) - C_{n-1}(x)|/|C_n(x)| \leq \tau \quad \text{oder} \quad |C_n(x)| \leq \tau$$

gilt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem  $x \in \mathbb{R}$  und  $\tau > 0$  eingelesen werden. Neben dem berechneten Wert  $C_n(x)$  sollen auch der korrekte Wert  $\cos(x)$  und der absolute Fehler  $|C_n(x) - \cos(x)|$  ausgegeben werden sowie der relative Fehler  $|C_n(x) - \cos(x)|/|\cos(x)|$  im Fall  $\cos(x) \neq 0$ . Schreiben Sie die Funktion möglichst so, dass diese mit einer Schleife auskommt und dass  $x^{2k}$  und  $(2k)!$  möglichst kostensparend realisiert werden. Man vermeide also insbesondere (vor- oder selbst implementierte) Funktionen zur Berechnung der Potenz oder der Faktoriellen. Speichern Sie den Source-Code unter `cos.c` in das Verzeichnis `serie07`.

**Aufgabe 7.3.** 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.

**Aufgabe 7.4.** Implementieren Sie folgendes Computerspiel. Der Computer merke sich eine zufällige Zahl zwischen 0 und 15. Sie haben maximal drei Versuche um die richtige Zahl zu erraten. Geben Sie beim ersten oder zweiten Versuch eine falsche Zahl an, soll der Computer Ihnen mitteilen, ob die angegebene Zahl größer oder kleiner als die gesuchte Zahl ist. Wenn man auch beim dritten Versuch daneben liegt, soll die richtige Zahl angezeigt werden. Zufallszahlen zwischen 0 und 15 können Sie folgendermaßen erzeugen: Zunächst binden Sie die Headerdateien `stdlib.h` und `time.h` in Ihr Programm ein. Danach können Sie in einer beliebigen Funktion mit

```
 srand( (unsigned) time(NULL) );  
 zufallszahl = (int) (16.0*rand()/(RAND_MAX+1.0));
```

eine Zufallszahl zwischen 0 und 15 generieren. Die Variable `zufallszahl` ist dabei vom Typ `int`. Außerdem soll die Möglichkeit bestehen das Spiel so oft wie gewollt zu wiederholen. Speichern Sie den Source-Code unter `spiel.c` in das Verzeichnis `serie07`.

**Aufgabe 7.5.** Schreiben Sie eine Funktion  $\text{kgV}(a,b)$ , die das kleinste gemeinsame Vielfache zweier natürlicher Zahlen  $a, b \in \mathbb{N}$  berechnet. Zur Lösung können Sie entweder die Primfaktoren beider Zahlen berechnen oder den Zusammenhang  $a \cdot b = \text{ggT}(a,b) \cdot \text{kgV}(a,b)$  berücksichtigen. Speichern Sie den Source-Code unter `kgv.c` in das Verzeichnis `serie07`.

**Aufgabe 7.6.** Welchen Output liefert das folgende Programm und warum?

```
#include <stdio.h>

void square(double* x)
{
    double* y;
    *y=(*x)*(*x);
}

int main(){
    double x=2.1;
    square(&x);
    printf("x^2=%f\n",x);
    return 0;
}
```

Verändern Sie das Programm geeignet, sodass der Output den Erwartungen entspricht.

**Aufgabe 7.7.** 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 `serie07`.

**Aufgabe 7.8.** Schreiben Sie eine Funktion `minmaxmean`, die von einem gegebenem Vektor  $x \in \mathbb{N}^n$  das Minimum, das Maximum und den Mittelwert  $\frac{1}{n} \sum_{j=1}^n x_j$  berechnet und geeignet zurückgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor  $x \in \mathbb{N}^n$  einliest und Minimum, Maximum und Mittelwert ausgibt. Die Länge  $n \in \mathbb{N}$  des Vektors soll eine Konstante im Hauptprogramm sein, die Funktion `minmaxmean` ist für beliebige Länge  $n$  zu programmieren. Speichern Sie den Source-Code unter `minmaxmean.c` in das Verzeichnis `serie07`.