

Übungen zur Vorlesung  
Einführung in das Programmieren für TM

Serie 4

Bibliotheken, While und For Schleife, Code testen

Die Pflichtübungen sind mit \* gekennzeichnet. Verwenden Sie Kommentare (`/*...*/` oder `//...`), um Ihren Code zu dokumentieren, wie im Skript gezeigt. Spezifizieren Sie in einem Blockkommentar (`/* ... */`) am Ende des Codes, wie Sie Ihre Implementierung getestet haben.

**Aufgabe 4.1.** \* 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 *nicht-rekursive* 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. Wie haben Sie Ihren Code auf Korrektheit getestet? Speichern Sie den Source-Code unter `goldenerSchnitt.c` in das Verzeichnis `serie04`.

**Aufgabe 4.2.** \* Schreiben Sie eine nicht-rekursive 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  $x \in \mathbb{R} \setminus \{0\}$  und für  $n < 0$  gilt  $x^n = (1/x)^{-n}$ . Weiters gilt  $0^n = 0$  für  $n > 0$ . Die Potenz  $0^n$  ist für  $n \leq 0$  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 4.3.** Schreiben Sie eine Funktion `double folgenglied(int n)`, die für gegebenes  $n \in \mathbb{N}$  das Folgenglied  $a_n := (-1)^n/(\sqrt{n} + 2)$  berechnet und zurückgibt. Schreiben Sie ferner ein Programm, das  $n \in \mathbb{N}$  von der Tastatur einliest, die Funktion aufruft und das Folgenglied  $a_n$  am Bildschirm ausgibt. Sie dürfen die Bibliothek `math.h` verwenden. Speichern Sie den Source-Code unter `folgenglied.c` in das Verzeichnis `serie04`.

**Aufgabe 4.4.** Schreiben Sie eine Funktion `int product(int x, int y)`, die für zwei gegebene natürliche Zahlen  $x$  und  $y$  das Produkt  $x * y$  berechnet und zurückgibt. Die Funktion darf nur die arithmetischen Operationen `+` und `-` verwenden. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem  $x$  und  $y$  eingelesen werden und das Ergebnis  $x * y$  ausgegeben wird. *Hinweis:* Verwenden Sie eine geeignete `while`-Schleife. Speichern Sie den Source-Code unter `produkt.c` in das Verzeichnis `serie04`.

**Aufgabe 4.5.** Schreiben Sie eine Funktion `double scanfpositive()`, die vom Benutzer die Eingabe einer positiven Zahl  $\tau > 0$  verlangt und diese dann zurückgibt. Die Eingabe soll solange wiederholt werden, bis die eingegebene Zahl  $\tau \in \mathbb{R}$  strikt positiv ist, d.h. bei Eingabe einer Zahl  $\tau \leq 0$  wird der Benutzer zu erneuter Eingabe aufgefordert. Schreibens Sie weiters ein aufrufendes Hauptprogramm. Speichern Sie den Source-Code unter `scanfpositive.c` in das Verzeichnis `serie04`.

**Aufgabe 4.6.** Schreiben Sie eine Funktion `void vielfache(int k, int nmax)`, die alle ganzzahligen Vielfachen der Zahl  $k \in \mathbb{N}$ , die  $\leq n_{\max} \in \mathbb{N}$  sind, 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 schreiben Sie ein Hauptprogramm, das die Daten  $k$  und  $n_{\max}$  von der Tastatur einliest und die Funktion aufruft. Speichern Sie den Source-Code unter `vielfache.c` in das Verzeichnis `serie04`.

**Aufgabe 4.7.** Ein Tripel  $(x, y, z) \in \mathbb{N}^3$  natürlicher Zahlen heißt *pythagoräisches Zahlentripel*, falls  $x^2 + y^2 = z^2$  gilt. Das wohl bekannteste Beispiel ist  $(3, 4, 5)$ . Offensichtlich gelten  $z > \max\{x, y\}$  sowie  $x \neq y$  und ohne Beschränkung der Allgemeinheit ferner  $x < y$ . Schreiben Sie eine `void`-Funktion `pythagoras`, die zu gegebener Schranke  $n \in \mathbb{N}$  alle pythagoräischen Zahlentripel mit  $x < y < z \leq n$  bestimmt und ausgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Schranke  $n$  eingelesen und `pythagoras` aufgerufen wird. Speichern Sie den Source-Code unter `pythagoras.c` in das Verzeichnis `serie04`.

**Aufgabe 4.8.** Schreiben Sie eine Funktion `double geometricMean(double x[])`, die von einem gegebenen Vektor  $x \in \mathbb{R}_{\geq 0}^3$  den geometrischen Mittelwert

$$\bar{x}_{\text{geom}} = \sqrt[3]{x_1 x_2 x_3}$$

berechnet und zurückgibt. Sie dürfen die Funktion `pow` aus der Bibliothek `math.h` verwenden. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das  $x \in \mathbb{R}^3$  über die Tastatur einliest und den geometrischen Mittelwert berechnet und ausgibt. Speichern Sie den Source-Code unter `geometricMean.c` in das Verzeichnis `serie04`.