

### 3. Funktionen

#### Lernziele:

Vertiefen der Kenntnisse über die Verwendung von Funktionen.

#### Aufgabe 1: Währungsumrechnung

- a) Schreiben Sie eine C-Funktion `euro2dollar`, welche einen übergebenen Euro-Betrag in Dollar umwandelt.

Hinweis: Derzeit (Stand: 27.10.2013) erhält man 1,3810095 Dollar für einen Euro.

- b) Schreiben Sie ferner eine C-Funktion `dollar2euro` zur Umwandlung von Dollar-Beträgen nach Euro.

- c) Schreiben Sie ein Hauptprogramm (Hauptfunktion `main`), welches vom Benutzer einen Geldbetrag einliest und diesen eingegebenen Betrag jeweils (als Euro-Wert interpretiert) in Dollar umrechnet sowie analog den Eingabebetrag (von Dollar) in Euro umwandelt.

Verwenden Sie zur Währungsumrechnung jeweils die obigen Funktionen.

Um diese Funktionen testen zu können, soll die Ausgabe mit maximaler Genauigkeit (d.h. ohne Rundung) erfolgen.

#### Aufgabe 2: Währungstabelle

- a) Schreiben Sie eine C-Funktion `printTable`, welche die folgenden Tabellen ausgibt.

Euro	Dollar	Dollar	Euro
-----+	-----	-----+	-----
100.00	138.10	100.00	72.41
200.00	276.20	200.00	144.82
300.00	414.30	300.00	217.23
40.000	552.40	400.00	289.64
500.00	690.50	500.00	362.06
600.00	828.61	600.00	434.46
700.00	966.71	700.00	506.88
800.00	1104.81	800.00	576.29
900.00	1242.91	900.00	651.70
1000.00	1381.01	1000.00	724.11

Anfangs- und Endwert sowie die Schrittweite für die Währungstabellen sollen an die Funktion `printTable` übergeben werden. (In den vorherigen Beispiel-Tabellen ist Anfangswert = 100.00, Endwert = 1000.00 und Schrittweite = 100.00 für die Umrechnungstabelle Euro nach Dollar sowie für Dollar nach Euro angenommen.)

Die umgerechneten Beträge sollen jeweils durch Aufrufe der Umrechnungsfunktionen aus Aufgabe 1 berechnet werden.

Hinweis: Beachten Sie die Formatierung der Ausgabe. So gilt u.a. 100 Cent = 1 Euro, ferner soll das Komma in jeder Zeile an derselben Stelle stehen.

- b) Schreiben Sie ferner ein Hauptprogramm, in dem der Nutzer Anfangs- und Endwert sowie die Schrittweite für die Währungstabellen eingeben kann.

### Aufgabe 3: Datum

- a) Schreiben Sie eine C-Funktion `monateImJahr`, welche die Anzahl der Monate eines Jahres zurückgibt.
- b) Schreiben Sie eine ferner eine C-Funktion `tageImMonat`, welche die Anzahl der Tage eines Monats zurückgibt.
- c) Schreiben Sie schließlich ein Hauptprogramm, in dem Sie ein Datum – d.h. 3 Zahlen für Tag, Monat und Jahr – einlesen und prüfen, ob das eingegebene Datum korrekt ist.

Beispiele:

- Tag = 19, Monat = 11, Jahr = 2013 → 19.11.2013 ist korrekt
- Tag = -6, Monat = 11, Jahr = 2013 → -6.11.2013 ist ungültig (Tag ungültig)
- Tag = 1, Monat = 17, Jahr = 7 → 1.17.7 ist ungültig (Monat ungültig)
- Tag = 31, Monat = 11, Jahr = 1900 → 31.11.1900 ist ungültig (Tag ist ungültig im Monat November)

### Testat:

Voraussetzung ist jeweils ein fehlerfreies, korrekt formatiertes C-Programm. Der korrekte Programmlauf muss anhand einer Beispieleingabe nachgewiesen werden. Sie müssen in der Lage sein, Ihr Programm im Detail zu erklären und ggf. auf Anweisung hin zu modifizieren.

**Nicht ausgelastet?** Dann haben wir hier noch einige optionale Zusatzaufgaben für Sie:

- Erweitern Sie das Programm in Aufgabe 3 dahingehend, dass Sie die Anzahl der Tage bis zum nächsten Weihnachtsfest (24.12.) vom eingegebenen Datum ermitteln.

Schreiben Sie dazu eine Funktion **tageBisWeihnachten**.

- Erweitern Sie das Programm in Aufgabe 3 dahingehend, dass Sie die Anzahl der Tage zwischen zwei beliebigen Daten ermittelt.

Schreiben Sie dazu eine passende Funktion, welche die Differenz an Tagen zwischen zwei beliebigen (korrekten) Daten berechnet.

Hinweise:

- Beachten Sie, dass die Daten sich auf evtl. auf verschiedene Jahre beziehen.
- Außerdem kann die Differenz negativ sein.

Beispiele:

- 19.11.2013 bis 20.11.2013 → 1 Tag
  - 19.11.2013 bis 18.11.2013 → -1 Tag
  - 19.11.2013 bis 19.12.2013 → 30 Tage
  - 19.11.2013 bis 24.12.2013 → 35 Tage
  - 19.11.2013 bis 19.11.2014 → 365 Tage
  - 19.11.2013 bis 24.12.2014 → 400 Tage
- Schreiben Sie ein Programm, welches das Datum des Ostersonntags zu einem gegebenen Jahr ermittelt und das Ergebnis wie folgt ausgibt:  
  

```
Ostern ist am 04. April im Jahr 2010
Ostern ist am 24. April im Jahr 2011
Ostern ist am 08. April im Jahr 2012
Ostern ist am 31. März im Jahr 2013
```
  - Verwenden Sie zur Osterberechnung eine C-Funktion, die zu einem übergebenen Jahr jeweils das Osterdatum zurückliefert. In der Berechnungsfunktion soll keine Ausgabe bzw. Ausgabeformatierung erfolgen!