

Prozedurale Programmierung – Übung 9

WS 2018/19 Prof. Dr. F.J. Schmitt

Hochschule **Rosenheim**
University of Applied Sciences



In der Community wird ein (fast leeres) Projekt in der Datei „eurotest.zip“ bereitgestellt.

Schreiben Sie ein C-Programm, das die Seriennummer einer Euro-Banknote auf Gültigkeit testet.

Die Seriennummer einer Euro-Banknote besteht aus folgenden Teilen:

- Buchstabe (Ländercode)
- 10-stellige Nummer
- Prüfziffer

Der Buchstabe am Anfang dient als Kennzeichnung für das Land, dessen Zentralbank die Banknote ausgegeben hat. Beispiel: P02571225193

P	Ländercode für Niederlande, Großbuchstabe A-Z
0257122519	10-stellige Nummer (nur Ziffern)
3	Prüfziffer

Die Prüfziffer ergibt sich wie folgt:

1. ersetze den Buchstaben des Ländercodes durch seinen ASCII-Code
Beispiel: P = 80
2. Bilde die Quersumme aus dem Ländercode und der 10-stelligen Nummer
Beispiel: $80 + 0 + 2 + 5 + 7 + 1 + 2 + 2 + 5 + 1 + 9 = 114$
3. Wähle die Prüfziffer so, dass die Gesamtsumme durch 9 teilbar ist
Beispiel: $3 \rightarrow 114 + 3 = 117 \quad (= 0 \text{ mod } 9)$

Das bedeutet: Bei einer gültigen Seriennummer ist die Gesamt-Quersumme inklusive Prüfziffer ohne Rest durch 9 teilbar.

Aufgabe 1

- a) Schreiben Sie zunächst eine Funktion `quersumme()`, die die Quersumme einer **12**-stelligen Seriennummer berechnet. Beachten Sie folgenden Randbedingungen:
 - die Seriennummer wird als Zeichenkette übergeben und darf von der Funktion nicht verändert werden
 - gehen Sie in der Funktion davon aus, dass die übergebene Zeichenkette tatsächlich eine gültige Seriennummer enthält (d.h.: der Aufbau ist korrekt, nur die Quersumme ist möglicherweise nicht ohne Rest durch 9 teilbar)
 - der Rückgabewert der Funktion ist die Quersumme als int
- b) Schreiben Sie in `main()` ein Testprogramm, das die Funktion mit mehreren Seriennummern aufruft und sowohl die Nummer als auch die Quersumme ausgibt.

Aufgabe 2

- a) Schreiben Sie nun eine weitere Funktion `eurotest()`, die eine 12-stellige Seriennummer auf Gültigkeit überprüft. Beachten Sie die folgenden Randbedingungen:
- die Seriennummer wird als Zeichenkette übergeben und darf von der Funktion nicht verändert werden
 - der Rückgabewert der Funktion ist vom Typ `t_errcode`, der wie folgt definiert ist:

```
typedef enum
{
    ec_ok,                /* Seriennummer gültig */
    ec_pz_falsch,         /* Prüfziffer falsch */
                          /* (also: Rest bei Division durch 9 ungleich Null)*/
    ec_zukurz,            /* Seriennummer zu kurz */
    ec_zulang,            /* Seriennummer zu lang */
    ec_LCfalsch,          /* Ländercode ist kein Großbuchstabe */
    ec_SNkeineZiffer       /* 10-stellige Seriennummer oder */
                          /* Prüfziffer enthält Zeichen, */
                          /* die nicht Ziffern sind */
} t_errcode;
```

- alle in diesem typedef genannten Fehler sollen auch geprüft werden
 - rufen Sie zur Berechnung der Quersumme die Funktion `quersumme()` aus (a) auf, sobald Sie sicher sind, dass die Zeichenkette gültig ist
- b) Erweitern Sie das Testprogramm in `main()` so, dass sowohl gültige Seriennummern als auch alle o.g. Fehlerfälle mindestens einmal geprüft werden. Geben Sie das Ergebnis der Prüfung auf dem Bildschirm aus. Schreiben Sie für die Ausgabe des Fehlercodes eine Funktion.

Aufgabe 3

Schreiben Sie eine Funktion `myprintf()`, der ein Formatstring und eine variable Anzahl von Argumenten übergeben wird. Der Formatstring soll aufgebaut und interpretiert werden wie zu der Funktion `printf()`.

Lediglich die Platzhalter sollen eingeschränkt sein auf `%d` und `%c`.