1 Testataufgabe: Sudoku

Das Ziel dieser Testataufgabe ist die Erstellung eines einfachen Sudoku-Spiels für die Konsole, wie die untere Abbildung darstellt. Bei einem Sudoku

> „[...] ist es das Ziel, ein 9×9-Gitter mit den Ziffern 1 bis 9 so zu füllen, dass jede Ziffer in jeder Spalte, in jeder Zeile und in jedem Block (3×3-Unterquadrat) genau einmal vorkommt – und natürlich in jedem der 81 Felder exakt eine Ziffer eingetragen wird. Ausgangspunkt ist ein Gitter, in dem bereits mehrere Ziffern vorgegeben sind.“[1]

```
    1  2  3   4  5  6   7  8  9
  +---------+---------+---------+
1 | 4  1    |    6  5 |       7 |
2 |       6 |       7 | 4  8    |
3 | 2     7 | 4  9    |       6 |
  +---------+---------+---------+
4 |    6    |    7    | 1       |
5 | 3     1 | 5       |    7  2 |
6 |    9    |    4  2 | 3     8 |
  +---------+---------+---------+
7 | 1     8 | 6       |    2  9 |
8 |    2    |    1  8 | 6  4    |
9 | 6       | 3       |    1    |
  +---------+---------+---------+

Eingabe:
```

1.1 Datenstruktur **(1 Punkt)**

Erstellen Sie ein globales, zweidimensionales und statisches Array[2] zur Speicherung des 9×9 Sudoku-Spielfelds. Ein leeres Feld kann z.B. durch die „0“ repräsentiert werden.

1.2 Bildschirmausgabe **(3 Punkte)**

Schreiben Sie eine Funktion, welche das Sudoku-Spielfeld ähnlich wie in oberer Abbildung übersichtlich auf dem Bildschirm ausgibt.

1.3 Dateneingabe **(4 Punkte)**

Erweitern Sie ihre `main`-Funktion, sodass der Benutzer Werte in ihrem Spielfeld ändern kann, bis das Sudoku vollständig ausgefüllt ist. Überlegen Sie sich, welche Eingaben Sie vom Benutzer benötigen, wie Sie die Benutzereingaben auf Gültigkeit überprüfen und wie der Benutzer „falsche Entscheidungen“ Rückgängig machen kann. Nutzen Sie dazu die `scanf`-Funktion[3] oder für zeichenweises Einlesen die `getchar`-Funktion.

[1] `https://de.wikipedia.org/wiki/Sudoku`
[2] `http://openbook.rheinwerk-verlag.de/c_von_a_bis_z/011_c_arrays_009.htm`
[3] `http://openbook.rheinwerk-verlag.de/c_von_a_bis_z/004_c_ein_ausgabe_001.htm`

1.4 Regelprüfung **(2 Punkte)**

Prüfen Sie nach einer Benutzereingabe, ob der Spielzug nach den drei Sudoku-Regeln erlaubt ist und weisen Sie den Benutzer auf eine mögliche Regelverletzung hin.

2 Präsenzaufgaben

1. Schreiben Sie ein Programm, das zwei eindimensionale `float`-Arrays `a,b` (Vektoren) fester Länge N definiert und initialisiert. Von diesen Vektoren soll das Skalarprodukt $s = \sum_{i=0}^{N-1} a[i] \cdot b[i]$ berechnet und auf dem Bildschirm ausgegeben werden. Dabei werde N im Rahmenprogramm als Makro definiert.

2. Schreiben Sie ein Programm, das ein eindimensionales `float`-Array fester Länge N definiert und initialisiert. Entweder das vom Betrag **größte** oder **kleinste** Element des Arrays soll auf dem Bildschirm ausgegeben werden.

3. Schreiben Sie ein Programm, das drei eindimensionale `float`-Arrays `a`, `b` und `c` fester Länge 3 definiert. `a` und `b` sollen geeignet initialisiert werden. In `c` soll das Kreuzprodukt berechnet und auf dem Bildschirm ausgegeben werden:

$$c = a \times b = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}.$$

4. Schreiben Sie ein Programm, in dem eine `int32_t`-Variable `x` deklariert wird und der Inhalt der auf `x` folgenden vier Bytes als `int`-Variable mit `printf` ausgegeben wird.

5. Schreiben Sie ein Programm zur Multiplikation von zwei Matrizen

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ 2 & 1 \end{pmatrix}.$$

   Die Ergebnismatrix $C \in \mathbb{R}^{2\times 2}$ soll auf dem Bildschirm ausgegeben werden.

6. Schreiben Sie ein Programm, in dem ein `int`-Array der Länge 5 definiert wird. Die Felder dieses Arrays sollen jeweils mit ihren zu `int` konvertierten Adressen gefüllt und ausgegeben werden. Was kann man an der Ausgabe sehen?

7. Schreiben Sie eine Funktion `cycle3`, die Zeiger auf drei `int`-Variablen `a,b,c` als Parameter besitzt und den Inhalt dieser Variablen zyklisch vertauscht.

   Beispiel: $a = 6, b = 4, c = 5$ sollen nach dem Funktionsaufruf die Werte $a = 5, b = 6, c = 4$ haben.