

4.5.A Datenstrukturen/Mehrdimensionale Arrays – Arbeitsblatt

Aufgabe 1: Datumsausgabe

Es soll ein Programm `datumaus.c` geschrieben werden, das nach Eingeben von Tag, Monat und Jahr das Datum wie im folgenden Beispiel ausgibt:

```
Terminal
schueler@debian964:~$ datumaus
Eingabe Tag : 24
Eingabe Monat: 12
Eingabe Jahr : 2012
=====
Es ist der 24. Dezember 2012.
```

Die Besonderheit liegt hier in der Ausgabe des Monats: Das Programm ermittelt aus der Zahl 12 die für den zwölften Monat passende Zeichenkette "Dezember". Dazu soll ab jetzt keine Mehrfachauswahl mehr verwendet werden. Stattdessen soll die Zuordnung durch ein Array von Zeichenketten passieren. Eine Zeichenkette ist selbst ein Array von `char`; also braucht man ein Array von Arrays von `char`.

- Warum reicht nicht ein einfaches Array von `char`?
- Wie viele Elemente hat das Array, und von wo bis wo (z.B. 1 bis 14) läuft der Index?
Anzahl der Elemente:

Der Index läuft von:

Der Index läuft bis:
- Schreiben Sie die Vereinbarung des Arrays:
- Mit welcher `printf()`-Zeile kann man das Element Nr. 3 des Arrays auf den Bildschirm ausgeben?
- Schreiben Sie das Programm!

Aufgabe 2: Ameisenzucht auf dem Bildschirm

Das Spiel `ameisenzucht.c` stellt eine einfache Simulation einer Pflanzen- oder Tierbevölkerung dar. Hier die Spielregeln:

- Das Spielfeld ist 20 Kästchen breit und 12 Kästchen hoch. Auf jedem dieser Kästchen kann entweder eine Ameise leben (z.B. durch ein `[x]` gekennzeichnet) oder nicht (z.B. durch einen Punkt `[.]` gekennzeichnet).
- Jedes Kästchen hat acht Nachbarn (links, rechts, oben, unten und die vier Diagonalen).
- Die Randkästchen oben grenzen an die Randkästchen unten, die Randkästchen links grenzen an die Randkästchen rechts, so dass — ähnlich wie auf einer Kugel — eine unbegrenzte Oberfläche entsteht.
- Zu Beginn kann man, wie man will, bestimmte Felder besetzen und andere Felder freilassen.
- Durch Druck auf die Return-Taste geht das Spiel in die nächste Runde (eine neue Ameisengeneration kann entstehen).
- Eine neue Ameise entsteht auf dann einem Feld, wenn mindestens zwei Nachbarfelder belegt sind (entsprechend zwei Ameisen-Eltern), aber maximal sechs Nachbarfelder belegt sind (sonst wird die Aufzucht zu schwierig).
- Sind alle acht Nachbarfelder eines Feldes belegt, dann wird es für eine dort schon lebende Ameise zu eng, und sie kann dort nicht mehr weiterleben. Sie stirbt dann, und ihr Feld wird frei.

Aufgabe 3: Fussball: Ergebnisse und Tabellen

Es liegt eine zweidimensionale Tabelle der Ergebnisse von Fußballspielen vor:

		Gast		
		DSC	SCP	VFL
Heim	DSC	—	3:1	1:2
	SCP	3:3	—	1:0
	VFL	1:1	0:1	—

Es handelt sich um eine kleine Liga mit nur drei Mannschaften. Daraus wurde ein zweidimensionales C-Array gemacht:

		x		
		0	1	2
y	0	0	3 - 1	1 - 2
	1	3 - 3	0	1 - 0
	2	1 - 1	0 - 1	0

Das Programm `fussball.c` soll nun aus diesen Spielergebnissen für jede Mannschaft die Punktzahl und die Tordifferenz berechnen. Es ist jedoch noch nicht fertig. Bisher existiert nur der Eingabeteil (hier als eine Folge von Zuweisungen) und der Ausgabeteil (Ausgabe der Punktzahlen und Tordifferenzen).

```

1 #include <stdio.h>
2 #define ANZ 3
3 int main(void)
4 {
5     /* y   x   */
6     /* Zl. Sp. */
7     int spiel[ANZ][ANZ]={{0,0,0},{0,0,0},{0,0,0}};
8     int punkte[ANZ]={0,0,0};
9     int tore[ANZ]={0,0,0};
10    int lauf, heim, gast;
11    spiel[0][1]=3-1; /* 3:1 */
12    spiel[0][2]=1-2; /* 1:2 */
13    spiel[1][0]=3-3; /* 3:3 */
14    spiel[1][2]=1-0; /* 1:0 */
15    spiel[2][0]=1-1; /* 1:1 */
16    spiel[2][1]=0-1; /* 0:1 */
17    printf("Erstellung_einer_Rangtabelle_aus_Spielergebnissen\n");
18    for(heim=0; heim<ANZ; ++heim)
19        for(gast=0; gast<ANZ; ++gast)
20            if(heim!=gast)
21                printf("Spiel_Mannsch.%d_gegen_Mannsch.%d:_Tordiff.%d\n",
22                    heim, gast, spiel[heim][gast]);
23    /* Punkteliste und Torliste ausrechnen: (fehlt noch) */
24    /* Tabellen ausgeben: */
25    for(lauf=0; lauf<ANZ; ++lauf)
26        printf("Mannschaft %d_hat %d_Punkte.\n", lauf, punkte[lauf]);
27    for(lauf=0; lauf<ANZ; ++lauf)
28        printf("Mannschaft %d_hat_Tordiff.:%d.\n", lauf, tore[lauf]);
29    return 0;
30 }

```

- Berechnen Sie die Tordifferenz für jede Mannschaft und schreiben Sie sie in das Array `tore`!
- Berechnen Sie die Punkte für jede Mannschaft und schreiben Sie sie in das Array `punkte`!