

## Aufgabe 9.1

Welche Ausgabe erzeugt das folgende Programm? Versuchen Sie zuerst die Antwort zu finden ohne das Programm auszuprobieren.

```
#include <stdio.h>
int main()
{
    int a[10] = {1,2,3,4};
    a[5] = 8;
    for (int i=0;i<10;i++) printf("a[%i]=%i\n",i,a[i]);
    return 0;
}
```

## Aufgabe 9.2

Eine von mehreren Möglichkeiten zum Erzeugen von Zufallszahlen in C ist die Funktion *rand()*. Diese Funktion liefert eine Zufallszahl im Bereich von 0-RAND\_MAX (incl.).

Zur Initialisierung des Zufallszahlengenerators muss ein mal die Funktion *srand()* mit einem Startwert als Parameter aufgerufen werden.

Füllen Sie in einer Schleife ein Array von 20 *int*-Elementen mit Zufallszahlen im Bereich von 0-1000 und geben Sie das so gefüllte Array in einer zweiten Schleife aus.

Wie verhalten sich die Zufallszahlen wenn Sie

- *srand()* weglassen
- *srand()* mit verschiedenen (festen) Übergabeparametern aufrufen
- *srand()* mit *time(NULL)* aus *time.h* als Parameter aufrufen
- *srand()* jedes mal vor *rand()* aufrufen?
- wie könnte eine Funktion *rand\_range(min,max)* aussehen, die Zufallszahlen im Bereich min...max erzeugt?

## Aufgabe 9.3

Ein einfaches Sortiervverfahren ist **Bubblesort**.

Dabei wird in einem Schleifendurchlauf jedes Element eines Arrays mit dem darauf folgenden Element verglichen. Wenn die beiden Elemente in der falschen Sortierreihenfolge liegen werden die Elemente vertauscht. Dieser Vorgang wird so lange wiederholt, bis ein vollständiger Durchlauf ohne Vertauschung durchgeführt wurde.

Erzeugen Sie mit dem Programm aus 9.2 ein Array mit 25 Zufallszahlen und sortieren Sie das Array anschließend mit Bubblesort. Geben Sie die sortierte Liste aus.