



Aufgabe 1. Implementiere eine Funktion, die den Wert zweier `int`-Variablen vertauscht.

Aufgabe 2. Erweitere die Funktion zur Lösung einer quadratischen Gleichung von gestern: Es soll möglich sein beide Lösungen weiter zu verwenden.

Aufgabe 3. Schreibe ein Modul `arrayhelpers`, das einige nützliche Funktion zum `int`-Array-Handling enthält:

- a) Array zeilenweise oder mit Kommata getrennt ausgeben
- b) Array sortieren
- c) Alle Felder eines Arrays auf einen einzelnen Wert setzen
- d) Array um 1 rotieren (d.h. das hinterste Element an erste Stelle schreiben und alle anderen Elemente um eins nach hinten schieben)
- e) Array um k rotieren
- f) Array umdrehen
- g) Ein Array in einem anderen suchen und die Position zurückgeben. Ist das Array nicht im anderen enthalten, so soll der Rückgabewert -1 sein.

```
1 int A[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
2 int B[3] = {4, 5, 6}  
3 int C[2] = {5, 7}  
4 int D[2] = {9, 10}
```

B ist an 3-ter Stelle in A enthalten und D an 8-ter. Das Array C ist garnicht in A enthalten, darum wird der Rückgabewert -1 sein.

Aufgabe 4.

- a) Schreibe eine Funktion, die prüft, ob ein Array eine Permutation ist. Ein Array mit n Elementen ist eine Permutation, wenn es die Zahlen von 0 bis $n - 1$ jeweils einmal enthält.
- b) Schreibe eine Funktion, die ein beliebiges `int`-Array und ein Permutationsarray entgegennimmt und das Array entsprechend permutiert. Das heißt: Wenn `a` ein Array und `p` ein Permutationsarray ist, dann soll nach Aufruf der Funktion `a[i]` an die Stelle `a[p[i]]` geschrieben worden sein.