



Aufgabe 1. Erweitere die Funktion zur Lösung einer quadratischen Gleichung von gestern: Es soll möglich sein beide Lösungen weiter zu verwenden.

Aufgabe 2. Implementiere eine Funktion, die den Inhalt zweier `int`-Variablen vertauscht.

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 mit einem Wert initialisieren
- 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 zurück geben. Sollte das Array nicht im anderen enthalten sein, so soll der Rückgabewert -1 sein.

Beispiel:

```
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}
```

Hier gilt: 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. Löse das N -Damen-Problem für eine Konstante N : Platziere N Damen auf einem Schachbrett so, dass sie sich paarweise nicht bedrohen.



Aufgabe 5. Wir wollen ein Array mit n Einträgen als Permutation interpretieren, wenn jede Zahl von 0 bis $n - 1$ darin vorkommt.

- a) Schreibe eine Funktion, die prüft, ob ein Array eine Permutation ist.
- b*) Schreibe eine Funktion, die ein Array und ein Permutationsarray entgegennimmt und das Array entsprechend permutiert.