



Aufgabe 1. 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 2. Wir sagen, ein Array der Länge n ist eine *Permutation*, wenn jede Zahl zwischen 1 und n an irgendeiner Stelle im Array vorkommt.

- a) Schreibe eine Funktion, die prüft, ob ein Array eine Permutation ist.
- b) Schreibe eine Funktion, die ein Array `a` und eine Permutation `p` entgegennimmt und `a` so umsortiert, dass das i -te Element an die Stelle `p[i]` geschrieben wird.

Aufgabe 3. Schreibe eine Funktion zur Lösung einer quadratischen Gleichung $aX^2 + cX + b = 0$, welche beide Lösungen berechnet und mit Hilfe von Pointern zurückgibt.



Aufgabe 4. a) Implementiere eine c-Datei zu folgender Header-Datei:

```
1  /* gibt die Länge eines Strings zurück */
2  int str_len(char *s);
3
4  /* gibt 0 zurück, wenn zwei strings gleich
5   * sind und 1 sonst */
6  int str_cmp(char *s1, char *s2);
7
8  /* kopiert s nach d und gibt d zurück */
9  char *str_cpy(char *d, char *s);
10
11 /* hänge s2 and s1 an und gib s1 zurück */
12 char *str_cat(char* s1, char* s2)
```

b) Implementiere nun noch folgende String-Funktionen, die man in der Praxis *niemals* benutzen würde:

```
1  /* schreibt s rückwärts in s und gib es zurück */
2  char *str_reverse(char *s);
3
4  /* gibt 1 zurück, wenn ein String ein Palindrom
5   * ist und 0 sonst */
6  int str_ispalin(char *s);
```