



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

Aufgabe 2. Schreibe Funktionen `square_to` und `root_to`, die einen `double`-Pointer entgegen nehmen, die dort stehende Variable quadrieren bzw. daraus die Wurzel ziehen und das Ergebnis sowohl zurück geben als auch an die gleiche Speicherstelle schreiben.

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.

Bevor du dir Stundenlang den Kopf über die Sternchen-Aufgabe zerbrichst, mache lieber Aufgabe 3.

Aufgabe 4. 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.



Aufgabe 5. 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);
```

und teste deinen Code mit folgendem Modul:

```
1 #include <stdio.h>
2 #include "mystrings.h"
3
4 int main() {
5     char p[100] = "Pepsi_";
6     char c[100] = "Coca_";
7     char suffix[10] = "Cola";
8     char out[100];
9     str_cpy(out,p);
10    str_cat(out,suffix);
11    str_cpy(p,out);
12    str_cpy(out,c);
13    str_cat(out,suffix);
14    str_cpy(c,out);
15    if (str_cmp(p,c)) {
16        printf("%s",p);
17        printf("_is_not_");
18        printf("%s",c);
19        printf("\n");
20    }
21    return 0;
22 }
```