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.

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

Aufgabe 2. Wir wolle ein Array mit n Einträgen als Permutation interpretieren, wenn jede Zahl von 0 bis $n-1$ darin vor kommt.

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 3. Implementiere eine c-Datei zu folgender Header-Datei:

```c
/* gibt die Länge eines Strings zurück */
int str_len(char *s);

/* gibt 0 zurück, wenn zwei strings gleich
 * sind und 1 sonst */
int str_cmp(char *s1, char *s2);

/* kopiert s nach d und gibt d zurück */
char *str_cpy(char *d, char *s);

/* hänge s2 and s1 an und gib s1 zurück */
char *str_cat(char* s1, char* s2)
```

und teste deinen Code mit folgendem Modul:

```c
#include <stdio.h>
#include "mystrings.h"

int main() {
        char p[100] = "Pepsi ";
        char c[100] = "Coca ";
        char suffix[10] = "Cola";
        char out[100];
        str_cpy(out,p);
        str_cat(out,suffix);
        str_cpy(p,out);
        str_cpy(out,c);
        str_cat(out,suffix);
        str_cpy(c,out);
        if (str_cmp(p,c)) {
                printf("%s",p);
                printf(" is not ");
                printf("%s",c);
                printf("\n");
        }
        return 0;
}
```