

Aufgabe 1. Das folgende Programm sollte ein Array mit n Zahlen allokalieren. In den i -ten Eintrag $1/(i+1)$ schreiben und das Array ausgeben. Dieses Programm benutzt die Funktion `printarray` aus dem Modul "arrayhelpers". Leider haben wir dabei 5 Fehler gemacht. Schnapp sie dir alle!

```
1  /* Array erstellen, mit 1/(i+1) fuellen und ausgeben.
2   * (c) 2015 Clelia und Johannes */
3
4  #include <stdio.h>
5  #include "arrayhelpers.h"
6
7  int main () {
8      double *array;
9      int n,i;
10
11     n = 42;
12     /* Hole Speicher fuer n Eintraege */
13     array = malloc (n);
14     if (array == NULL) {
15         printf ("Fehler, nicht genug Speicher.\n");
16     }
17     for (i = 0; i < sizeof (array); i++) {
18         /* Schreibe 1/(i+1) in Array */
19         array[i] = 1/(i+1);
20     }
21     printarray (array, n); /* Gebe Array aus */
22     free (array);
23     return 0;
24 }
```

Aufgabe 2. a) Schreibe ein Programm, welches eine Datei im folgenden Format ausliest: In der ersten Zeile steht die Anzahl der folgenden Zeilen. Die Zeilen sehen dann so aus:

```
1  cos(0.4) = 0.921
2  cos(0.45) = 0.9014
3  cos(0.6) = 0.82533
```

Wenn in einer Zeile das Ergebnis um mehr als 10^{-3} von dem tatsächlichen Wert abweicht, so soll diese Differenz mit Zeilennummer auf der Konsole ausgegeben werden.

- b) * Modifiziere das Programm so, dass in der ersten Zeile nicht mehr die Anzahl der Einträge stehen muss.

Aufgabe 3. Diese Aufgabe läuft auf die Implementierung des Merge-Sort Algorithmus hinaus.

- a) Implementiere eine Funktion `merge`, die zwei bereits sortierte (eventuell verschieden große) Arrays als Argumente erhält, diese zu einem sortierten Array kombiniert und dieses zurück liefert.
- b) Die Funktion `mergesort` selbst soll ein Array in zwei (möglichst gleich große) Teilarrays zerlegen, sich für diese Teilarrays selbst aufrufen und danach die dann sortierten Teilarrays mit der `merge`-Funktion kombinieren. Erhält die Funktion ein Array mit keinem oder einem Element so belässt es dieses Array wie es ist, dann ist es nämlich bereits sortiert.
- c) Schreibe ein Programm, dass einen Dateinamen auf der Kommandozeile entgegen nimmt, den Inhalt der Datei in ein `int`-Array einliest, dieses sortiert und es dann wieder in die gleiche Datei zurück schreibt.

Hier als Tipp ein Vorschlag für die Signaturen der beiden Funktionen:

```
1 int *merge(int *list1, int n, int *list2, int m);  
2 void mergesort(int *list, int n);
```