Aufgabe 1. In dieser Aufgabe geht es um numerische Integration.

a) Implementiere eine Integrationsfunktion, die das Intervall $[a, b]$ in n gleich große Teile aufteilt, für diese jeweils die Trapzsumme (aus der Vorlesung) berechnet und diese aufsummiert:

```
1 double integrate(double a, double b,
2   double (*f)(double), unsigned int n);
```

b) Schreibe nun eine Funktion, die nicht die Anzahl der Teilintervalle erhält, sondern eine "Fehlertoleranz" e. Die Funktion die Aufteilung solange verfeinern, bis sich der approximierte Wert für das Integral durch eine Verfeinerung nur noch um weniger als e ändern würde.

Aufgabe 2. Implementiere eine Funktion die zu einem gegebenen Funktionenpointer $f : \mathbb{R} \to \mathbb{R}$, einem Dateinamen, einer Schrittweite $s \in \mathbb{R}$, einer Startstelle x_1 und einer Endstelle x_2 die Wertetabelle der Funktion zwischen x_1 und x_2 zur Schrittweite s speichert. Dabei sollen x und $f(x)$ durch einen Tabulator getrennt werden und jedes Paar $(x, f(x))$ in einer eigenen Zeile stehen. Etwa wäre die Ausgabe für $f = \cos$ zwischen $x_1 = 0$ und $x_2 = 0$ mit Schrittweite $s = 0.1$ die folgende:

```
1  0.0 1.0
2  0.1 0.995004165278
3  0.2 0.980066577841
4  0.3 0.955336489126
5  0.4 0.921060994003
6  0.5 0.87758256189
7  0.6 0.82533561491
8  0.7 0.764842187284
9  0.8 0.696706709347
10 0.9 0.621609968271
11 1.0 0.540302305868
```

FLIP ME

Aufgabe 3. Auf der Homepage gibt es ein Modul `rationals`, in dem rationale Zahlen implementiert sind. Binde es in ein neues Projekt ein und deklariere ein Array von Brüchen, in etwa so:

```
RATIONAL arr[7] = {
 {-1,2}, {1,2}, {3,4}, {9,7}, {10,1}, {7,3}, {11,2} };
```

Nun sortiere `arr` mit Hilfe von `qsort`.

Aufgabe 4. Schreibe eine Funktion, die mithilfe von `qsort` ein Array von Strings lexikographisch (wie im Telefonbuch) sortiert. Das heißt, dass erst nach der ersten Stelle sortiert wird, dann nach der Zweiten usw. In der `<string.h>` liegt eine Vergleichsfunktion für diese Situation vor: `strcmp`. Als Beispiel hier eine lexikographisch sortierte Liste:

```
1
10
133
2
2344
Hallo
Thor
Tor
```