Aufgabe 1. Implementiere eine Funktion die zu einem gegebenen Funktionenpointer $f : \mathbb{R} \to \mathbb{R}$, einen 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.

Aufgabe 2. 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 Trapezsumme (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 3. Sortiere ein Array deiner `rationals` mithilfe 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 1
2 10
3 133
4 2
5 2344
6 Hallo
7 Thor
8 Tor
```

Aufgabe 5. Schreibe ein Programm, dass ein Labyrinth aus einer Datei einliest:

```
XXXXXXXXXXXXXXXX
X  X  XXXXXXXXXX*X
X$X  XX      XXX  X
X  X  XX  XXX  XXX  X
X     XX  XXX  XXX  X
XXX  X    XX  XXX  X
XXX    X          X
XXXXXXXXXXXXXXXX
```

Bemerkung: Wir spezifizieren das Labyrinth hier nicht viel näher, entscheide dich selbst vorher was für ein Format die Datei haben soll und welche Einschränkungen du daran stellst: Soll die Größe des Labyrinths variabel sein oder fest? Soll die Größe in der ersten Zeile der Datei stehen oder nicht? Soll das Labyrinth quadratisch sein oder nicht? Soll es außen herum immer mit Xen begrenzt sein oder hast du vielleicht eine andere Lösung?
Das Programm soll einen Weg vom Startpunkt (dem Stern, dem Geburtsort) zum Dollar (dem Schatz) finden. Die Xe sind Wände und Leerzeichen sind Pfade. Markiere einen Weg mit Punkten und gebe das Labyrinth mit Weg in der Konsole aus.

```
XXXXXXXXXXXXXXXX
X  X  XXXXXXXXXX*X
X$X  XX      XXX.X
X.X  XX  XXX  XXX.X
X...XX  XXX  XXX.X
XXX.X...XX  XXX.X
XXX...X........X
XXXXXXXXXXXXXXXX
```