



Aufgabe 1. Implementiere eine Funktion die zu einem gegebenen Funktionspointer $f : \mathbb{R} \rightarrow \mathbb{R}$, 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(  
2     double a,  
3     double b,  
4     double (*f)(double),  
5     unsigned int n  
6 );
```

- *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. 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 4. Brainfuck ist eine sogenannte esoterische Programmiersprache, das sind Sprachen, die meist zu wissenschaftlichen oder theoretischen Zwecken, oder einfach zum Spaß entwickelt wurden. Sbesteht nur aus 8 Befehlen: > < + - , . [] alle anderen Zeichen werden als Kommentar interpretiert. Diese Befehle werden, wie bei C auch, nacheinander ausgeführt. Sie operieren auf einem (potentiell unendlich langen) Band (welches aus Zellen besteht in denen jeweils ein char steht) indem sie einen Lese-/Schreibkopf über das Band bewegen und Zeichen lesen / schreiben lassen. Das Band ist überall mit '\0' vorinitialisiert und der Lese-/Schreibkopf startet an "Position 0" des Bandes. Die Befehle haben folgende Bedeutung:

> bzw. <	schiebt den Lese-/Schreibkopf eins nach rechts bzw. links
+ bzw. -	in- bzw. dekrementiert den Bandwert unter dem Lese-/Schreibkopf um 1
.	gibt den Wert unter dem Lese-/Schreibkopf aus
,	liest ein Zeichen vom Benutzer ein und schreibt es unter den Lese-/Schreibkopf
[springt zum zugehörigen]-Befehl, wenn der Wert unter dem Lese-/Schreibkopf 0 ist, sonst soll nichts passieren
]	springt zum zugehörigen [-Befehl, wenn der Wert unter dem Lese-/Schreibkopf verschieden von 0 ist

So sieht ein "Hallo-Welt"-Programm in Brainfuck aus:

```

1 ++++++
2 [
3   >++++++>++++++>++++>+<<<<-
4 ]
5 >++.
6 >+.
7 ++++++.
8 +++.>+.
9 <<+++++.
10 >.++.
11 - - - - - . - - - - - .
12 >+.>.

```

Deine Aufgabe ist es nun, ein Programm zu schreiben, welches Brainfuck-Programme einlesen und ausführen kann.