

**Aufgabe 1.** Implementiere die Addition, Multiplikation und Division sowie das Kürzen rationaler Zahlen in einem eigenen Modul.

**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.** Schreibe ein Modul, das Vektorrechnung auf  $\mathbb{R}^n$  implementiert. Vektoren sollen **structs** von einem **double**-Array und dem **int**  $n$  sein. Implementiere die nachfolgenden Funktionen:

- a) eine Funktion, die genügend Speicher für einen Vektor reserviert und einen Pointer darauf zurück gibt
- b) Vektoraddition
- c) Vektorsubtraktion
- d) Produkt eines Vektors mit einer skalaren Größe
- e) Skalarprodukt zweier Vektoren
- f) Kreuzprodukt zweier Vektoren (falls existent)
- g) eine Funktion, die prüft, ob zwei Vektoren orthogonal zueinander stehen
- h) eine Funktion, die prüft, ob zwei Vektoren parallel zueinander sind
- i) eine Funktion, die einen Vektor auf der Konsole aus gibt

```
1 void make0 (VEKTOR *a) {  
2     int i;  
3     for(i = 0; i < a->n; i++) a[i] = 0;  
4 }
```