

Prozedurale Programmierung – Übung 5

WS 2020/21

Prof. Dr. F.J. Schmitt

Hochschule **Rosenheim**
University of Applied Sciences



In der Community wird ein (fast leeres) Projekt in der Datei „bruchrechnung.zip“ bereitgestellt.

Aufgabe 1

Wie Sie wissen stellt C keinen elementaren Datentyp für rationale Zahlen zur Verfügung. Daher sollen Sie nun einen solchen neuen Datentyp sowie entsprechende Funktionen zur Bruchrechnung bereitstellen. Alle Funktionsdeklarationen sowie Definitionen von Datentypen sollen in `mainfile.h` stehen.

Definieren Sie zunächst einen neuen Datentyp `bruch_s`, der Zähler und Nenner eines Bruchs enthält.

Schreiben Sie anschließend vier verschiedene Funktionen, die

- zwei Brüche als Parameter übergeben bekommen
- Produkt, Quotient, Summe bzw. Differenz der zwei übergebenen Brüche berechnen
- und einen Bruch als Ergebnis zurückliefern

Nehmen Sie folgende Vereinfachungen an:

- die Brüche müssen nach der Berechnung **nicht** gekürzt werden
- verwenden Sie als Hauptnenner bei der Summen- bzw. Differenzberechnung das Produkt der Nenner der beiden Brüche (was evtl. größer ist als nötig)

Schreiben Sie ein Programm (innerhalb von `main`), das von der Tastatur zwei Brüche (Zähler und Nenner getrennt) einliest, die vier Funktionen aufruft und die Berechnungsergebnisse ausgibt. Testen Sie Ihr Programm mit verschiedenen Brüchen. Die Ausgabe sollte wie folgt aussehen:

```
Bitte Zaehler und Nenner Bruch 1 eingeben: 1 3
Bitte Zaehler und Nenner Bruch 2 eingeben: -2 3
1/3 * -2/3 = -2/9
1/3 / -2/3 = 3/-6
1/3 + -2/3 = -3/9
1/3 - -2/3 = 9/9
```

Aufgabe 2 – Zusatzaufgabe für Fortgeschrittene

Erweitern Sie das Programm so, dass in den vier Funktionen die berechneten Brüche vor der Rückgabe so weit wie möglich gekürzt werden. Testen Sie Ihr Programm mit verschiedenen Brüchen.

Die Ausgabe sollte dann wie folgt aussehen:

```
Bitte Zaehler und Nenner Bruch 1 eingeben: 1 3
Bitte Zaehler und Nenner Bruch 2 eingeben: -2 3
1/3 * -2/3 = -2/9
1/3 / -2/3 = 1/-2
1/3 + -2/3 = -1/3
1/3 - -2/3 = 1/1
```