Programmieren für Physiker

Interfakultatives Institut für Anwendungen der Informatik
Institut für Theoretische Teilchenphysik

Prof. Dr. M. Steinhauser, Dr. A. Mildenberger
`http://comp.physik.kit.edu`

SS 2011 – Blatt 11
Bearbeitungszeitraum: bis 29. Juni 2011

Klausurtermin: Dienstag, 12. Juli 2011, 17.30 Uhr.
Hörsaal gemäß erstem Buchstaben des Nachnamens: A-R $\widehat{=}$ Gerthsen-HS,
S-Z $\widehat{=}$ Gaede-HS.
Dauer: 90 Minuten. Es sind keine Hilfsmittel erlaubt. Bitte Studentenausweis mitbringen.
Bachelor-Studierende: Es ist erforderlich, sich bis 05.07.2011 unter https://studium.kit.edu/ zur Klausur anzumelden.

Aufgabe 27: rationale Zahlen **Pflichtaufgabe**

Programmieren Sie in C++ eine Klasse `class Ratio { ... }`, die das Rechnen mit Brüchen ermöglicht. Zähler und Nenner sollen von außen unzugängliche `long` Variablen sein. Implementieren Sie folgende Methoden:

- Eine Konstruktor mit zwei Argumenten, um Zähler und Nenner einen Wert zuzuweisen.
- Kürzen (mittels Berechnung des größten gemeinsamen Teilers, siehe hierzu entweder Vorlesung oder Euklidischer Algorithmus).
- Addition zweier Brüche
- Subtraktion zweier Brüche
- Multiplikation zweier Brüche
- Division zweier Brüche
- Unäres Minus (d.h. Minus als Vorzeichen)
- Ausgabe in der Form *Zähler/Nenner*
- Berechnung des Gleitkommawerts (`double`) des Bruchs

Überladen Sie den Ausgabeoperator `<<` so, dass Ratio-Variablen direkt innerhalb eines `cout` in der Form *Zähler/Nenner* ausgegeben werden.
Bei allen Rechenoperationen soll nach der eigentlichen Berechnung das Ergebnis sofort vollständig gekürzt werden. Verwenden Sie die Operatoren `+ - * /`. Achten Sie darauf, dass auch die Rechnung mit negativen Brüchen richtig funktioniert.
Berechnen Sie damit im Hauptprogramm

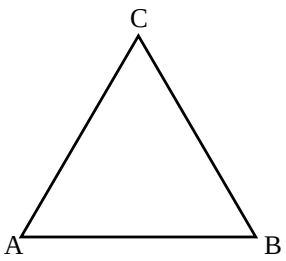
$$\frac{2}{25}\Big/\frac{7}{5}-\frac{2}{5}\cdot\left(-\frac{1}{4}+\frac{1}{3}\right)$$

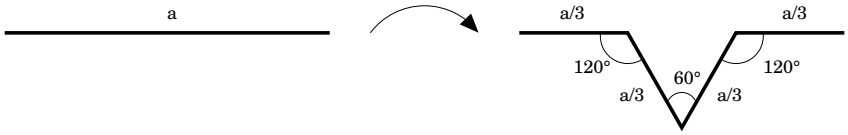
und geben Sie den Wert als Bruch und als Gleitkommazahl aus. Die einzelnen Brüche dürfen hierbei explizit im Hauptprogramm eingetragen sein.

Aufgabe 28: Kochsche Schneeflocke **freiwilliges Zusatztestat**[1]

In dieser Aufgabe soll mit C++ eine grafische Ausgabe der Kochschen Schneeflocke programmiert werden. Auf der WWW-Seite zur Vorlesung finden Sie eine Anleitung zur Verwendung der Grafikbibliothek `CImg` (`http://cimg.sourceforge.net/`) unter Linux und Windows.

Die Kochsche Schneeflocke ist ein fraktales Objekt, das durch folgende Vorschrift erzeugt wird: Man beginnt mit einem gleichseitigen Dreieck ABC, siehe rechte Abbildung. Jede Seite des Dreiecks, also die Strecken $\overline{AB}$, $\overline{BC}$ und $\overline{CA}$, wird nun mit der unten skizzierten Substitutionsregel ersetzt. Bei der Substitution wird eine Strecke in drei gleich lange Abschnitte zerlegt. An Stelle des mittleren Abschnitts tritt eine nach außen orientierte „Zacke“, die aus Teilen eines dreifach kleineren, gleichseitigen Dreiecks gebildet wird.





Jede einzelne der dabei neu entstandenen Strecken wird nun wieder mittels der Substitutionsregel ersetzt. Der Vorgang wird insgesamt so oft wiederholt, bis schließlich eine vorgegebene Ersetzungstiefe n erreicht wird. Geben Sie für eine einzulesende Ersetzungstiefe $0 \leq n < 10$ die Figur grafisch auf dem Bildschirm aus.

Zusatzfragen: Welchen Umfang und welchen Flächeninhalt hat die Schneeflocke als Funktion von n? Was passiert für $n \to \infty$?

Schlüsselwörter in C++ bzw. C

`and`	`and_eq`	`asm`	`auto`•	`bitand`	`bitor`
`bool`	`break`•	`case`•	`catch`	`char`•	`class`
`compl`	`const`•	`const_cast`	`continue`•	`default`•	`delete`
`do`•	`double`•	`dynamic_cast`	`else`•	`enum`•	`explicit`
`export`	`extern`•	`false`	`float`•	`for`•	`friend`
`goto`•	`if`•	`inline`	`int`•	`long`•	`mutable`
`namespace`	`new`	`not`	`not_eq`	`operator`	`or`
`or_eq`	`private`	`protected`	`public`	`register`•	`reinterpret_cast`
`return`•	`short`•	`signed`•	`sizeof`•	`static`•	`static_cast`
`struct`•	`switch`•	`template`	`this`	`throw`	`true`
`try`	`typedef`•	`typeid`	`typename`	`union`•	`unsigned`•
`using`	`virtual`	`void`•	`volatile`•	`wchar_t`	`while`•
`xor`	`xor_eq`				

Schlüsselwörter können nicht für eigene Bezeichnungen verwendet werden.
Die Untermenge der C Schlüsselwörter (nach ANSI Standard) ist mit • markiert.

---

[1] Sie können bei dieser freiwilligen Aufgabe ein Testat erhalten, welches gewertet wird; die Aufgabe zählt aber nicht als Pflichtaufgabe.