

**Allgemeine Hinweise:**

- Bitte geben Sie möglichst in Gruppen **zu zwei** Leuten ab.

**Übung 1** *Rechengesetze in Fließkommazahlen*

Welche der folgenden Behauptungen sind wahr:

- a) Die Fließkommaoperation auf  $\mathbb{F}$  sind im allgemeinen kommutativ.
- b) Für die Addition auf  $\mathbb{F}$  gilt das Assoziativgesetz.
- c) Zwischen Multiplikation und Addition auf  $\mathbb{F}$  gilt das Distributivgesetz.

Widerlegen Sie die falschen Aussagen (geben Sie einfach jeweils ein Gegenbeispiel an), die richtige Aussagen müssen Sie nicht beweisen. **( 3 Punkte )**

**Übung 2** *Kondition der Standardoperationen*

Berechnen Sie die relativen Konditionszahlen (Verstärkungsfaktoren) der Standardoperationen *Multiplizieren*, *Dividieren* und *Wurzel ziehen*, also der Abbildungen:

$$F(x, y) = x \cdot y, \quad F(x, y) = \frac{x}{y} \quad F(x) = \sqrt{x}.$$

**( 3 Punkte )**

**Übung 3** *Landau Symbole*

Schreiben Sie die folgenden Ausdrücke in der Form  $f(h) = \mathcal{O}(h^m)$  (für  $h \rightarrow 0, h > 0$ ) mit einem möglichst großen  $m \in \mathbb{N}$ .

$$f(h) = (ph^2 + h)^2 - p^2h^4 \quad p \in \mathbb{N}$$

$$f(h) = -\frac{h^2}{\ln(h)}$$

$$f(h) = \frac{\sin(h)}{h} - 1;$$

$$f(h) = \frac{\sin(x+h) - 2\sin(x) + \sin(x-h)}{h^2} + \sin(x).$$

und skizzieren Sie jeweils  $f(h)$  zusammen mit dem jeweiligen  $c \cdot h^m$  auf dem Intervall  $(0, 1)$ .  
(Hinweis: Für einen der Ausdrücke ist die Form  $f(h) = o(h^m)$  vorzuziehen! )

**( 4 Punkte )**

**Übung 4** *Rundungsfehler positiver Subtraktion*

Gegeben seien drei positive Zahlen  $x_1, x_2$  und  $x_3$  mit  $x_1 \gg x_2, x_3$ . Welcher der beiden Algorithmen

$$(A1) \quad ((x_1 \ominus x_2) \ominus x_3)$$

$$(A2) \quad (x_1 \ominus (x_2 \oplus x_3))$$

zur Berechnung von  $x_1 - x_2 - x_3$  hat geringere Rundungsfehler im Ergebnis?

**( 2 Punkte )**

### Übung 5 Numerische Nulladdition die Zweite (Praktische Übung)

In der letzten praktischen Aufgabe sollten Sie durch manuelle Eingaben die größte positive Fließkommazahl  $x$  vom Typ `float` bzw. `double` bestimmen, für welche der Ausdruck

$$1.0 == x + 1.0$$

als wahr ausgewertet wird.

Schreiben Sie nun ein Programm `nulladdition`, welches diese Zahl unter Anwendung von Schleifen und bedingten Anweisungen für einen gegebenen Fließkommatyp automatisch findet. Die Benutzereingabe soll über die Kommandozeile

```
./nulladdition <Datentyp>
```

möglich sein, d.h. der Benutzer des Programms `nulladdition` gibt für `<Datentyp>` entweder `double` oder `float` ein.

Hinweise:

- Lagern Sie die Implementierung des Algorithmus in eine separate C++-Funktion aus. Um die Funktion nicht doppelt (also für `float` und für `double`) schreiben zu müssen, bietet es sich hier an, ein Template-Parameter (z.B. `REAL`) für den Datentyp zu verwenden, also:

```
template<typename REAL>
REAL get_limit()
{ ... }
```

Die Funktion `get_limit<Datentyp>()` muss dann mit dem passenden Datentyp explizit aufrufen werden.

- In der Zusammenfassung "Grundlegende Anweisungen in C++" (siehe Homepage der Vorlesung) ist ein Abschnitt über Kommandozeilenparameter aufgeführt.

- Benutzen Sie die Hauptfunktion

```
int main( int argc, char **argv )
{ ... }
```

- Überprüfen Sie, ob die Anzahl der Kommandozeilenargumente (`argc`) mindestens 2 ist.
- Wenn nicht geben Sie z.B. die Meldung aus

```
Gebrauch: ./nulladdition <double|float>
Beispiel: ./nulladdition float
```

und verlasse das Programm mit `exit(0);`

- Ansonsten lesen Sie den ersten Parameter als String ein:

```
std::string s1 = argv[1];
```

- Mit der Anweisung

```
if( s1 == "float" )
{ ... }
```

kann man überprüfen, ob der erste Parameter `float` heißt oder nicht.

( 6 Punkte )