

## 4. Aufgabenblatt

Abgabe 23.05.14

*Hinweis zu Programmieraufgaben:* Bitte machen Sie zu jeder Programmieraufgabe Testläufe, die die Funktionalität Ihrer Programme ausreichend dokumentieren. Die Testläufe und den Quellcode drucken Sie bitte aus, und geben beides mit der jeweiligen Übung auf Papier ab. Der Quellcode muss zusätzlich in Form einer assemblierbaren Datei in das Repository <https://svn.imp.fu-berlin.de/cst-teaching-schmitt/SS14/TI-2/Tut-<NUM>/grp<NUM>> eingereicht werden. Ihr jeweiliger Tutor wird Ihnen Näheres mitteilen.

### Problem 1: Zahlendarstellung – Rechnen

- Rechnen Sie folgende Zahlen (Basis 10) in eine 8-Bit Zweierkomplement-Darstellung um: -42, 121, -0
- Stellen Sie die Zahl  $00111001010110101_2$  im Dualzahlssystem in den Basen 4, 8, 10 und 16 dar.
- Rechnen Sie folgende Zahlen in das Hexadezimalsystem um. Die Genauigkeit soll 4 Nachkommastellen betragen.  $274.137_{10}$ ,  $10468.8765_{10}$ ,  $1243.3421_{10}$
- Berechnen Sie die normalisierten Ergebnisse.  $1.0101 \cdot 2^3 + 1.1001 \cdot 2^4$ ,  $1.1101 \cdot 2^3 \cdot 1.1101 \cdot 2^4$
- Stellen Sie  $-43.3652_{10}$  und  $105.8523_{10}$  in normalisierter Gleitkommadarstellung dar (1 Bit Vorzeichen, 4 Bit Charakteristik, 11 Bit Mantisse).

### Problem 2: Zahlendarstellung – Fragen

- Auf Grund der begrenzten Anzahl verfügbarer Bits pro Wort muss ein Kompromiss zwischen der Größe der Mantisse und des Exponenten gefunden werden. Was bedeutet dies im Grunde? Nennen Sie zur Verdeutlichung jeweils einen Fall in denen eine der beiden Komponenten möglichst groß/klein gewählt werden sollte.
- Was muss beim Verwenden von Gleitkommazahlen stets bedacht werden? Nennen Sie mindestens drei verschiedene Punkte.

### Problem 3: Collatz Conjecture

Gegeben sei das nebenstehende Stück Pseudo-Code. Implementieren Sie diese Funktion in einem x86-Assembler-Programm! Testläufe und evtl. Bildschirmausgaben dürfen gerne in einem C-Wrapper implementiert werden.

Hinweis:  $\text{is-even}(x)$  ist äquivalent zu “ $x$  modulo 2 ist gleich 0”

```
function collatz(k)
  i <- 0
  while k > 1
    i <- i + 1
    if is-even(k)
      k <- k / 2
    else
      k <- k * 3
      k <- k + 1
  return i
```