

## Übungsblatt 3: Programmieren in C (WS 2018/19)

Abgabefrist Montag, 12.11.18, 12:00

1. Zur Beantwortung von Fragen und Hilfe bei Problemen stehen wir Ihnen immer in der Praktische Übung, Mittwoch 15:30, Raum 32-410-PC zur Verfügung sowie per Email an [pinc-support@cs.uni-kl.de](mailto:pinc-support@cs.uni-kl.de)!
2. Sie können sich im Exclaim System unter <https://softtech.cs.uni-kl.de/exclaim> mit Ihrem persönlichen STATS-Account einloggen und Dateien zu den einzelnen Übungen hochladen.
3. Die Aufgaben auf diesem Übungsblatt sind alle freiwillig zu bearbeiten.

### 1 Kleinste Zahl (Abgabe in Exclaim als `min3.c`)

Erstellen Sie ein C-Programm, das drei ganze Zahlen einliest und die kleinste dieser Zahlen ausgibt.

1. Geben Sie einen Algorithmus zur Berechnung der kleinsten Zahl zunächst als Flussdiagramm an.
2. Implementieren Sie Ihren Algorithmus als C-Programm `min3.c`.

Die Ausgabe des Programms soll folgendermaßen gestaltet sein:

```
Das Minimum von 5, -4, 7 ist -4.
```

### 2 Summe berechnen (Abgabe in Exclaim als `summe.c`)

Erstellen Sie ein C-Programm, das eine positive ganze Zahl  $n$  einliest und die Zahlen  $1, 2, \dots, n$  aufaddiert.

1. Geben Sie einen Algorithmus zur Berechnung der Summe zunächst als Flussdiagramm an.
2. Implementieren Sie Ihren Algorithmus als C-Programm `summe.c`.

Die Ausgabe des Programms soll bei Eingabe 5 folgendermaßen gestaltet sein:

```
Die Summe von 1 bis 5 ist 15.
```

### 3 Binäre Zahlendarstellung (keine Abgabe)

1. Ergänzen Sie folgende Tabelle, sodass jede Zeile den gleichen Zahlenwert in dem jeweiligen Zahlenformat hat.

Bitmuster(8B)	Int	Unsign.Int.	Hex	Okt	ASCII
10010110	127	179	FB	64	U

2. Rechnen Sie die folgende Zahlen direkt - ohne über Dezimalzahlen zu gehen dafür aber über Binärzahlen - in die rechts des Zuweisungspfeils angegebenen Zahlenformate (Basen) um.
  - $1001001010111111_2 \Rightarrow \text{_____}_8$
  - $1001001010111111_2 \Rightarrow \text{_____}_{16}$
  - $CBAF_{16} \Rightarrow \text{_____}_8$
  - $726_8 \Rightarrow \text{_____}_{16}$
3. Welche Zahlen sind hier als single-precision float-Zahlen dargestellt?
  - 01000000111000000000000000000000
  - 11000011010110000000000000000000
4. Stellen Sie die folgenden Zahlen als single-precision float-Zahlen dar!
  - $178,625_{10}$
  - $10010110_2$
  - $-152,8_{10}$
5. Lösen Sie die folgenden Rechenaufgaben in Binärarithmetik (Binär-Addition, d.h. keine Subtraktion)!
  - $17 + 19$
  - $16 - 10$
  - $18 - 23$
  - $(-8) - (-11)$
6. Falls nötig, berechnen Sie hierzu zunächst das Zweierkomplement der gegebenen Dezimalzahlen.
  - $17 + 19$
  - $16 - 10$
  - $18 - 23$
  - $(-8) - (-11)$

## 4 Wie teuer ist das Laden eines Smartphone-Akkus? (Abgabe in Exclaim als `smartphone.c`)

Die meisten Smartphone-Akkus halten gerade mal einen Tag durch. Deshalb ist der Griff zum Ladegerät am Abend für viele Smartphone-Nutzer schon Routine. Aber was kostet es, wenn wir unser Smartphone jeden Tag aufladen?

Nehmen Sie hierzu an, dass Ihr Smartphone-Akku eine Kapazität von 3000 mAh bei einer Spannung von 3.85 V aufweist und täglich zu 80% geladen werden muss. Die Verluste des Ladegeräts können vernachlässigt werden. Eine kWh-Stunde Strom kostet im Durchschnitt 30 Cent.

1. Geben Sie die mathematische Formel an, um die Kosten für das Laden des Smartphones pro Tag zu berechnen.
2. Implementieren Sie Ihre Formel als C-Programm `smartphone.c`. Das Programm nimmt als Eingabe die Anzahl an Tagen, für die die Kosten berechnet werden sollen.

Die Ausgabe des Programms soll bei Eingabe 365 folgendermaßen gestaltet sein:

Kosten: 1.0118 EUR

**Hinweis:** Die Ausgabe des Geldbetrags muss mit 4 Nachkommastellen erfolgen.