

# Aufgaben zu Grundlagen der Informatik I

Steffen Oesterwind

## Übung 1

Ziel dieser Übung ist es, eine lauffähige Entwicklungsumgebung aus Editor und Compiler zu haben, sodass du für die folgenden Übungen vorbereitet bist.

### Aufgabe 1.1

Erkläre kurz die unterschiedlichen Aufgaben von Compiler und Linker.

### Aufgabe 1.2

Gebe das unten stehende Programm in einen Editor ein, kompiliere es und führe es aus.

```
1  #include <stdio.h>
2
3  int main(void) {
4      printf("Hello World!\n");
5      return 0;
6  }
```

### Aufgabe 1.3

Gebe das unterstehende Programm in einen Editor ein, kompiliere es und führe es aus.

```
1  #include <stdio.h>
2
3  int main(void) {
4      printf("Hello World: %d\n", 6*7);
5      return 0;
6  }
```

## Übung 2

Ziel dieser Übung ist es, den Umgang mit Variablen und Eingabe/Ausgabe zu lernen.

### Aufgabe 2.1

Wie groß ist der Wertebereich der Datentypen `char`, `unsigned char`, `int` und `unsigned int`. Was ist das jeweilige Verhalten bei Überlauf?

### Aufgabe 2.2

Schreibe ein Programm, welches eine Zahl vom Typ `int` mit `scanf` einliest und die Zahl selber sowie deren Quadrat mit `printf` ausgibt. Wie verhält sich die Ausgabe negativer Zahlen, wenn du die eingegebene Zahl als nicht vorzeichenbehaftete Zahl ausgibst?

### Aufgabe 2.3

Schreibe ein Programm, welches zwei Zahlen vom Typ `int` mit `scanf` einliest und beide Zahlen sowie deren Produkt und Summe mit `printf` ausgibt.

### Aufgabe 2.4

Schreibe ein Programm, welches zwei Zahlen vom Typ `int` mit `scanf` einliest und beide Zahlen sowie deren Quotient und Divisionsrest mit `printf` ausgibt.

### Aufgabe 2.5

Schreibe ein Programm, welches eine Zahl vom Typ `int` mit `scanf` einliest und die drei niedrigwertigsten Stellen in jeweils einer eigenen Zeile ausgibt. Abschließend soll der eventuell verbleibende Rest ausgegeben werden.

## Übung 3

### Aufgabe 3.1

Schreibe ein Programm, welches eine Temperatur in Fahrenheit als Gleitkommazahl einliest und diese in Celsius und Kelvin ausgibt. Ist die Temperatur unter 0° Celsius, soll „Winter“, zwischen 20° und 25° „Raumtemperatur“ und bei über 30° „Hitzefrei“ ausgegeben werden.

### Aufgabe 3.2

Schreibe ein Programm, welches eine einstellige Zahl einliest und die dazu gehörige Farbe gemäß den Farbcodes auf Widerständen ausgibt. Du kannst dir dafür einen der beiden Farbringe aussuchen. Ist die eingegebene Zahl ungültig, soll eine Fehlermeldung ausgegeben werden.

### Aufgabe 3.3

Schreibe ein Programm, welches zwei Uhrzeiten mit jeweils Stunde und Minute einliest, und den Absolutwert der Differenz in Stunden und Minuten sowie, ob die erste oder zweite Uhrzeit später am Tag war, ausgibt. Außerdem soll die erste Uhrzeit im 12h-Format mit AM/PM ausgegeben werden.

### Aufgabe 3.4

Schreibe ein Programm, welches zunächst eine Zahl einliest und diese zur Auswahl eines Operators verwendet: z.B. 1 für Addition, 2. für Inkrement. Darauf werden entweder eine oder zwei weitere Zahlen eingelesen und der Operator darauf angewandt, sowie das Ergebnis der Operation ausgegeben. Implementiere dies für fünf Operatoren deiner Wahl mit jeweils einem oder zwei Operanden.

### Aufgabe 3.5

Ein Busunternehmen vermietet Busse mit 50 Sitzplätzen zu 1000 Euro pro Tag. Entwickle ein Programm, das nach Eingabe einer Reisegruppengröße die folgenden Werte berechnet und ausgibt:

- Anzahl der nötigen Busse
- Anzahl der unbenutzten Plätze
- Durchschnittliche Auslastung eines Busses
- Reisepreis pro Person

Beispiel: Eine 120-Personen-Gruppe benötigt drei Busse, es sind 30 Plätze unbelegt, die Busauslastung beträgt 80 % und der Reisepreis pro Kopf 25 Euro.

## Übung 4

### Aufgabe 4.1 ...zunächst nur bis 4.4 bearbeiten!

Schreibe ein Programm, welches eine quadratische Gleichung der Form  $ax^2 + bx + c = 0$  für gegebenes  $a$ ,  $b$  und  $c$  auflöst. Existiert keine Lösung (negative Zahl unter der Wurzel) soll ein Fehler ausgegeben werden.

### Aufgabe 4.2

Schreibe ein Programm, welches Polarkoordinaten in kartesische umrechnet und umgekehrt.

### Aufgabe 4.3

Erweitere das Programm aus Afg. 3.2 so, dass es die Farbcodes für gegebenen Widerstandswert mit Einerstelle, Zehnerstelle und Multiplikator ausgibt. Dabei soll außerdem die Eingabe auf Gültigkeit überprüft und so lange wiederholt werden, bis der Anwender eine gültige Zahl eingibt. Gliedere das Programm in geeignete Funktionen, wobei die Ausgabe auch in einer Funktion geschehen kann.

### Aufgabe 4.4

Erweitere das Programm aus Afg. 2.5 so, dass es für eine beliebige Anzahl von Stellen verwendbar ist. Hier ist auch wie in Afg. 3.2 die Eingabe zu überprüfen: erlaubt sind nur positive Ganzzahlen!

### Aufgabe 4.5 ...ab hier erst nach Wiederholung von Bitoperationen!

Schreibe ein Programm, welches eine Zahl vom Typ `unsigned int` einliest und die 16 niedrigwertigsten Bits in binärer Darstellung ausgibt. Dazu sollen Bitoperationen verwendet werden. Die Ausgabe soll in einer Funktion implementiert sein, welche neben der eingegebenen Zahl auch die Anzahl auszugebender Stellen als Parameter erhält, und neben den Stellen (0/1) auch die Bitposition ausgeben soll, z.B. so:

```
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
1  0  1  1  0  0 1 0 1 1 1 0 1 0 0 1
```

### Aufgabe 4.6

Implementiere eine Funktion in einer separaten Datei, die beliebig große Dezimalzahlen vom Typ `unsigned int` in hexadezimale Darstellung umwandelt und teste diese in einem Programm mit interaktiver Eingabe. Implementiere dazu die Hilfsfunktion `char digit2hex(int dig)`, die zu einer Zahl die zugehörige Ziffer im Hexadezimalsystem zurück gibt (z.B. `digit2hex(10) → 'A'`). Implementiere diese Funktion ohne Fallunterscheidung für alle 16 möglichen Werte. Tipp: ASCII-Tabelle verwenden.

### Aufgabe 4.7

Implementiere eine Funktion, die eine Binärzahl in dezimaler Darstellung (z.B. 1101, also Eintausend Einhundert und Eins) in den korrekten Zahlenwert (13 in diesem Fall) umwandelt und teste diese in einem Programm mit interaktiver Eingabe. Verwende dazu Bitoperationen.

### Aufgabe 4.8

Implementiere eine Funktion, die in einer Zahl vom Typ `unsigned int` die gesetzten Bits zählt und teste diese in einem Programm mit interaktiver Eingabe. Zur einfacheren Kontrolle der Ausgabe soll die Zahl wie in Afg. 4.5 binär ausgegeben werden. Kopiere dazu die in Afg. 4.5 erstellte Funktion.

## Übung 5

### Aufgabe 5.1

Schreibe ein Programm, welches eine durch die Arraygröße festgelegte Anzahl an Zahlen vom Typ `int` in ein Array einliest und anschließend alle Zahlen, sowie deren Summe ausgibt.

### Aufgabe 5.2

Schreibe ein Programm, welches eine durch die Arraygröße festgelegte Anzahl an Zahlen vom Typ `int` in ein Array einliest und ausgibt, ob diese Zahlen aufsteigend sortiert sind.

### Aufgabe 5.3

Schreibe ein Programm, welches eine durch die Arraygröße festgelegte Anzahl an Zahlen vom Typ `int` in ein Array einliest und die Differenzen aufeinander folgender Elemente in ein weiteres Array schreibt und dieses ausgibt.

## Übung 6

### Aufgabe 6.1

Schreibe ein Programm wie in Afg. 5.3, implementiere dabei jedoch die Eingabe, Berechnung und Ausgabe in einer jeweils separaten Funktion.

### Aufgabe 6.2

Schreibe ein Programm wie in Afg. 4.2. Implementiere dabei die Berechnungen als eigene Funktionen.

### Aufgabe 6.3

Schreibe ein Programm, welches einen String von festgelegter Maximallänge einliest und bei Buchstaben die Groß-/Kleinschreibung vertauscht (`a → A`, `H → h`).

## Aufgabe 6.4

Schreibe ein Programm, welches eine Zeichenkette einliest und diese in eine Zahl vom Typ `unsigned int` konvertiert ("1337" → 1337). Ist ein ungültiges Zeichen (keine Ziffer) in der Eingabe enthalten, soll eine Fehlermeldung ausgegeben werden. Welche Möglichkeiten zur Kategorisierung der Zeichen existieren? (Tipp: C-Standardbibliothek verwenden)

Die Konvertierung ist in einer eigenen Funktion zu implementieren, die zurück gibt, ob die Konvertierung erfolgreich war.

## Übung 7

### Aufgabe 7.1

Schreibe ein Programm, welches einen String von festgelegter Maximallänge einliest und diesen String umdreht: "Hallo" → "ollaH". Implementiere das Umkehren des Strings in einer eigenen Funktion mit der Signatur: `void strreverse(char *s);`

### Aufgabe 7.2

Schreibe ein Programm, welches einen String von festgelegter Maximallänge einliest und erkennt, ob es sich bei dem eingegebenen String um ein Palindrom handelt (rückwärts gleich wie vorwärts zu lesen, z.B. Anna, Otto, neben). Dabei soll Groß/Kleinschreibung ignoriert werden.

### Aufgabe 7.3

Implementiere das Programm aus 4.3 durch Verwendung eines konstanten Arrays anstatt switch/case. z.B. so: `const char *farben[] = {...};`

## Übung 8

### Aufgabe 8.1

Deklariere eine Aufzählung (`enum`) `color_t` der Farben Rot, Lila, Gelb, Blau und Grau. Implementiere eine Funktion, die eine Farbe (`enum`) in eine dazugehörige Zeichenkette (`string`, z.B. „Rot“) umwandelt, sowie eine Funktion, die aus einer gegebenen Zeichenkette das passende Element der Aufzählung auswählt. Ist die gegebene Zeichenkette keiner Farbe zuzuordnen, soll Grau zurückgegeben werden. Die Signaturen der zu implementierenden Funktionen sind `const char *color_to_string(color_t color);` und `color_t color_from_string(const char *s);`

Lese zum Testen eine Farbe (`string`) ein, wandle sie in die zugehörige Aufzählung (`color_t`) um, und konvertiere sie anschließend wieder in einen `string`. Dieser String soll ausgegeben werden und wenn eingegebener und ausgegebener String übereinstimmen, funktionieren die Funktionen korrekt.

Tipp: Speichere die Zuordnung von Enum zu zugehörigem String für beide Funktionen in einer Lookup-Tabelle (Array eines `struct`, welches jeweils ein Enum und einen String enthält).

### Aufgabe 8.2

Deklariere eine Datenstruktur `vec2_t` für einen 2D-Vektor, wobei dessen Komponenten vom Typ `float` sein sollen. Implementiere zunächst Funktionen, die einen Vektor einlesen, bzw. ausgeben. Implementiere darauf basierend Funktionen, die Betrag, bzw. Winkel eines Vektors bestimmen, zwei Vektoren addieren/subtrahieren und das Skalarprodukt zweier Vektoren berechnen. Teste diese Funktionen in einem Programm mit interaktiver Eingabe.