Institut für Informatik
Prof. Dr. Stefan Conrad

Sommersemester 2020
Übung 1

Hardwarenahe Programmierung
Gruppe 14 (Lars)

In dieser Übung decken wir die Grundlagen der C-Programmierung ab, insbesondere Funktionen, Arrays, Pointer und Unit-Tests.

Wichtig:

- *Sie müssen **mindestens 11 von insgesamt 14 Pflichtaufgaben mit Tests lösen**, um die praktische Übung zu bestehen. Auf den Übungsblättern 1 bis 6 gibt es jeweils 2, 3, 2, 2, 3 und 2 Aufgaben mit Tests, die Sie lösen sollen. Die Tests erkennen Sie an den `*.ts`- oder `test.sh`-Dateien.*
- *Das **Abschlussprojekt muss vollständig gelöst werden**. Hier darf kein einziger Test fehlschlagen! Das Abschlussprojekt finden Sie auf Übungsblatt 7.*
- *Sie müssen **alle 6 Tests zu den Übungen im ILIAS** bestehen.*
- *Um besser auf die Abschlussaufgabe vorbereitet zu sein, sollten Sie sich aber bemühen, auch die optionalen Vorbereitungsaufgaben zu lösen.*
- *Denken Sie daran, dass Sie Ihre **Lösungen im ILIAS abgeben. Ihre Dateien müssen genau in der Form, in der sie hochgeladen wurden, kompilierbar sein**. Packen Sie dazu den vollständigen `uploads`-Ordner in eine Zip-Datei. Andere Formate sind nicht erlaubt. Die gegebenen Dateien dürfen nicht umbenannt und die Tests dürfen nicht verändert werden.*
- *Denken Sie daran, **Test 1 im ILIAS zu lösen**.*
- *Denken Sie daran, Ihre Strings zu terminieren und Ihre Variablen zu initialisieren!*
- *Nutzen Sie das Lernmodul, die vorgeschlagenen Bücher und das Internet, um sich auf die Übungsblätter vorzubereiten.*
- *Beginnen Sie frühzeitig mit den Übungsaufgaben. Wenn Sie zu spät anfangen, haben Sie nicht mehr die Möglichkeit, in Ihrer Übungsgruppe fragen zu stellen.*

Aufgabe 1 *Funktionen (Vorbereitungsaufgabe)*

(a) Erstellen Sie eine Funktion `max(int a, int b)`, die das Maximum der beiden Zahlen ausgibt.

(b) Schreiben Sie ein Programm, das mit Ihrer Funktion aus (a) zwei Zahlen x und y testet und dem Benutzer am Bildschirm eine Meldung ausgibt, welche der Zahlen kleiner ist. Dabei soll der Benutzer nach Programmstart aufgefordert werden, die beiden Zahlen einzugeben. Beispiel:

```
Geben Sie die erste Zahl ein:      12
Geben Sie die zweite Zahl ein:     13
-----------------------------------
Das Maximum von beiden Zahlen ist: 13
```

Aufgabe 2 *Pointer und Arrays (Vorbereitungsaufgabe)*

(a) Schreiben Sie ein Programm, das 10 Messwerte (jeweils vom Typ `float`) von der Konsole in ein Array einliest und den Inhalt des Arrays anschließend ausgibt.

(b) Falls Sie in Ihrer Lösung eckige Klammern, also `[]`, für die Lese- und Schreibzugriffe auf das Array verwendet haben, schreiben Sie Ihr Programm so um, dass es stattdessen Pointerzugriffe verwendet! Verwenden Sie eckige Klammern hier also **nur** noch zur Deklaration des Arrays.

Aufgabe 3 *Pointer und Funktionen (Vorbereitungsaufgabe)*

Schreiben Sie ein Programm, in dem zwei `int` Zahlen von der Tastatur eingelesen und in lokalen Variablen gespeichert werden. Die Werte der Variablen sollen anschließend in einer Funktion namens `swap` miteinander vertauscht werden. Geben Sie die Werte der beiden Variablen vor und nach dem Aufruf von `swap` auf dem Bildschirm aus.

Aufgabe 4 *Unittests (**Pflichtaufgabe**)*

In dieser Aufgabe sollen Sie die Funktionalität von vorgegebenen C-Funktionen mittels des Unit-Test Frameworks *check* überprüfen. Schauen Sie sich dazu zunächst die vorgegebenen Unittests in der Datei `simpleString_tests.ts` zur Implementierung in `simpleString.c` an, welche ihnen zeigen, wie solche Tests aussehen können.

(a) Bauen Sie die Tests und führen Sie sie aus (entsprechend der Anleitung im ILIAS-Lernmodul). Die fehlschlagenden Tests helfen Ihnen, Fehler in der von uns bereitgestellten Implementierung zu erkennen. Sie sollten verstehen, was die Ausgabe der Tests bedeutet.

(b) Korrigieren Sie im Anschluss die Fehler in der Implementierung und demonstrieren Sie, dass nun alle Tests erfolgreich durchlaufen. Die Tests selbst sollten Sie nicht verändern!

Folgende C-Funktionen sind gestellt:

- eine Funktion `int my_strcmp(char *string1, char *string2)`, die 1 zurückgeben soll, wenn die beiden übergebenen Strings gleich sind, und 0 sonst.
- eine Funktion `int count_in_string(char *haystack, char needle)`, welche die Anzahl zurückgeben soll, wie oft `needle` in `haystack` endhalten ist.

Aufgabe 5 *Pointer und Funktionen (**Pflichtaufgabe**)*

In dieser Aufgabe sollen Sie ein Programm schreiben, das einen Kaffeeautomaten repräsentiert. In der hochgeladenen Datei *kaffee.c* finden Sie einige vorgegebene Funktionsprototypen. Mit Ausnahme der `print_kaffee`-Funktion sind diese noch nicht implementiert.

Ein Kaffee kostet 1 Euro und der Automat verfügt zu Beginn über 100 Kaffees.

Das Programm soll wie folgt ablaufen:

1. Der Automat fragt den Benutzer, wie viele Kaffees gekauft werden sollen. Sie dürfen davon ausgehen, dass nur ganze Zahlen eingegeben werden.
2. Danach verringert der Automat die Anzahl der Kaffees und erhöht den Geldstand entsprechend. Außerdem gibt der Automat die gekauften Kaffees einzeln mit der Funktion `print_kaffee` und die Anzahl der noch verbleibenden Kaffees aus. Für den Fall, dass der Benutzer mehr Kaffees anfragt, als übrig sind, wird die Anzahl auf die noch übrige Menge beschränkt.
3. Wenn die Anzahl der angefragten Kaffees negativ ist, soll sich der Automat beenden.
4. Wenn danach die Geldkassette mit 50 Euro EUR gefüllt ist, gibt der Automat zusätzlich die Meldung `Kasse leeren` aus.
5. Wenn keine Kaffees mehr verfügbar sind, beendet sich der Automat. Sonst kehrt er zu Schritt 1 zurück.
6. Überprüfen Sie die Korrektheit ihres Programms, indem sie die Script-Datei `test.sh` ausführen. Eventuell müssen Sie die Datei mit dem Befehl `chmod +x test.sh` erst ausführbar machen.