1 Testataufgaben

1.1 Projekt (Woche 1) **(4 Punkte)**

In den letzten acht Wochen haben Sie die Grundlagen der prozeduralen Programmierung in der Programmiersprache `c` gelernt und schon viele kleinere Programme entwickelt. Nun ist es an der Zeit, sich an einem eigenen größeren Programm zu versuchen. Ihre Aufgabe ist es, in den Gruppen innerhalb von vier Wochen ein Projekt zu planen, umzusetzen und zu testen. Was Ihr Programm tun soll, ist Ihnen überlassen, egal ob CD-Verwaltung oder Spiel oder etwas vollkommen anderes. Die Aufgabenverteilung für die vier Wochen sieht wie folgt aus:

Woche 1	Planung des Projekts	(4 Punkte)
Woche 2, 3	Implementierung	(13 Punkte)
Woche 4	Test und Präsentation	(10 Punkte)

Begleitet wird das Ganze von kleineren Übungsaufgaben zur Vorlesung. Innerhalb der nächsten Woche soll die schriftliche Planung des Projekts erfolgen, **die Sie Ihrem Tutor im Testat spätestens am 10.01.2019 abgeben**. In der Planung sollen enthalten sein:

(A) **Spezifikation**

Eine ausführliche Beschreibung der Funktionalität. Nach der Fertigstellung werden Sie danach bewertet, inwieweit die Implementierung mit dieser Beschreibung übereinstimmt. Stellen Sie dafür mehrere (ca. 8) wesentliche Leistungsmerkmale auf, nach denen Ihr Programm bewertet werden soll.

(B) **Design**

Eine detaillierte Beschreibung der von Ihnen angestrebten technischen Umsetzung der in (A) spezifizierten Funktionalitäten:

(a) Programmstruktur/Aufbau des Programms

(b) wichtigste Funktionen und ihre Signatur (Parameter und Rückgabewert)

(c) zentrale (strukturierte) Datentypen

1.2 Übungsaufgabe zur Vorlesung **(3 Punkte)**

Definieren Sie eine Struktur namens `struct student`, in der entsprechende Studenteninformationen wie Vor- und Nachname, Matrikelnummer, Adresse und die Anzahl der von ihm bereits belegten Pflichtkurse abgelegt werden können.
Schreiben Sie dazu ein Programm, in welchem Sie ein Feld aus drei Elementen, mit folgender Datenbelegung erzeugen:

```
{"Anna", "Musterfrau", 22222, "Am Schwarzenberg-Campus 3", 4}
{"Hans", "Peter",      44444, "Kasernenstrasse 12",        2}
{"Lisa", "Lustig",     66666, "Denickestrasse 15",         8}
```

Das Programm soll das Feld in einer Schleife durchlaufen und deren Inhalt auf dem Bildschirm ausgeben. Anschließend soll das erste und letzte Element vertauscht und der Inhalt erneut ausgegeben werden.

1.3 Wiederholungsfragen **(3 Punkte)**

Nach der Wiederholung der Vorlesungsinhalte und der Übungsblätter 01 bis 09 sollte Ihnen die Beantwortung nachfolgender Fragen sehr leicht fallen. Ihr Tutor wird jedem Gruppenmitglied drei beliebige dieser Fragen stellen:

- Welche ganzzahligen Datentypen gibt es in C?
- Was sind die Gleitpunkt-Datentypen in C?
- Wie werden ganze Zahlen im Zweierkomplement kodiert?
- Worin unterscheiden sich `signed` und `unsigned` Integerzahlen?
- Was ist ein *„wrap around“* und wann tritt er auf?
- Wie wird eine `float`-Zahl in IEEE 754 kodiert?
- Wie heißt der Zeichen-Datentyp in C, was sind ASCII-Zeichen?
- Wie werden in C Zeichenketten (Strings) realisiert?
- Wie wird eine Funktion in C deklariert (Rückgabetyp, Eingabeargumente, ...)?
- Welche Kontrollfluss-Anweisungen in C kennen Sie bereits (Übersichtstabelle)?
- Was sind Arrays, wie werden sie definiert und wozu dienen sie?
- Welche Möglichkeiten kennen Sie, um ein Array zu initialisieren?
- Was ist ein Zeiger, wie wird er in C deklariert und was ist sein Datentyp?
- Welcher Zusammenhang besteht zwischen Zeigern und Arrays?
- Was passiert bei: `int a[2] = {1,2}, *pa = a; pa = pa + 1; *pa = 0;` ?
- Geben Sie zwei Möglichkeiten an, um auf das n-te Element des Arrays `a` zuzugreifen!
- Was sind dynamische Arrays, wie werden sie erzeugt und was ist dabei zu beachten?
- Was ist der Unterschied zwischen *Call-by-Value* und *Call-by-Reference*?
- Geben Sie ein kurzes Beispiel für *Call-by-Value* und *Call-by-Reference*?
- Was sind die Vor- und Nachteile von rekursiven und nicht rekursiven Funktionen?
- Schreiben Sie eine rekursive Funktion, welche die Fakultät einer Zahl n zurück gibt?
- Was ist die Aufgabe des Präprozessors, wann wird er aufgerufen und von wem?
- Welche Präprozessor-Anweisungen kennen Sie und was ist deren Bedeutung?
- Wie wird die `scanf()`-Funktion aufgerufen (kleines Beispiel)?
- Worin unterscheiden sich die Funktionen `fscanf`, `scanf` und `sscanf`?

2 Präsenzaufgaben

1. Schreiben Sie ein Programm, das einen String `str_in` von der Tastatur einliest und einen zweiten String `str_out` doppelter Länge erzeugt. In `str_out` sollen alle Buchstaben von `str_in` verdoppelt stehen und `str_out` soll auf dem Bildschirm ausgegeben werden.

   Beispiel: `str_in = "String"`, `str_out = "SSttrriinngg"`.

2. Schreiben Sie ein Programm, das die ersten 10 Wörter einer Datei in ein Array einliest und anschließend zeilenweise auf dem Bildschirm ausgibt.

3. Schreiben Sie ein Programm `add`, das in der Kommandozeile eingegebene Zahlen addiert und auf dem Bildschirm ausgibt.

   Beispiel: Der Aufruf `add.exe 1 2 3 4` erzeugt die Ausgabe „10“.

4. Schreiben Sie ein Programm `potenziere`, das zu zwei in der Kommandozeile angegeben (hinreichend kleinen) natürlichen Zahlen a, b die Potenz a^b berechnet und dieses Ergebnis in einer Datei ablegt, deren Name ebenfalls in der Kommandozeile angegeben ist.

   Beispiel: Der Aufruf `potenziere.exe 2 3 MyOut.txt` speichert das Ergebnis „8“ in einer Datei mit dem Namen „MyOut.txt“.

5. Definieren Sie eine Struktur „Student“ mit den Komponenten „Name“, „Vorname“, „Studiengang“, „Matrikelnummer“ und „FachSemester“. Überlegen Sie sich geeignete Datentypen für die einzelnen Strukturkomponenten. Die Komponente„Name“ soll ein `char`-Array fester Länge n sein. Welchen Wert hat n, wenn Namen mit maximal 20 Buchstaben gespeichert werden sollen?

6. Definieren sie eine Struktur für die Verwaltung von Kontodaten mit den Komponenten „Kontoinhaber“, „Kontonummer“ und „Bankleitzahl“. Überlegen Sie sich geeignete Datentypen für die einzelnen Strukturkomponenten. Schreiben Sie ein Hauptprogramm, was die Daten für ein Strukturelement von der Tastatur einliest.

7. Schreiben Sie ein Programm `mycopy`, das eine Datei kopiert. Dabei soll der Name der zu kopierenden Datei als auch der Name der Kopie als Kommandozeilenparameter übergeben werden.

   Beispiel: `mycopy.exe Original.txt Kopie.txt`

8. Schreiben Sie eine Funktion `stringcompare`, die als Eingabeparameter zwei Strings besitzt, diese vergleicht und als Ergebnis „1“ zurückliefert, falls beide identisch sind und andernfalls „0“.

9. Schreiben Sie eine Funktion `filecompare`, die als Eingabeparameter zwei Dateinamen besitzt, beide Dateien vergleicht und als Ergebnis „1“ zurückliefert, falls beide identisch sind und andernfalls „0“.