

Prozedurale Programmierung, Übungsblatt 09
letzter Abgabetermin 12. Januar 2017

1. Projekt

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 12. Januar 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.

(2 Punkte)

(b) **Design**

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

- i. Programmstruktur/Aufbau des Programms
- ii. wichtigste Funktionen und ihre Signatur (Parameter und Rückgabewert)
- iii. zentrale (strukturierte) Datentypen

(2 Punkte)

2. Übungsaufgabe zur Vorlesung

Definieren Sie eine Struktur namens `struct student`, in der entsprechende Studentinformationen 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.

(3 Punkte)

3. Am 12. Januar 2017 findet in der Hörsaalübung eine Probeklausur statt. Sie dient lediglich zur Überprüfung des eigenen Wissensstandes in Sachen c-Programmierung und wird nicht bewertet. Zur Vorbereitung ist es ratsam, den bisherigen Vorlesungsstoff und die Inhalte der Übungsblätter 01 bis 09 gut zu wiederholen. Die Beantwortung nachfolgender Fragen sollte Ihnen dann sehr leicht fallen. Ihr Tutor wird jedem Gruppenmitglied drei beliebige dieser Fragen stellen:

(3 Punkte)

- 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 (Rückgabotyp, 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)?
- Finden Sie heraus, worin sich die Funktionen `fscanf`, `scanf` und `sscanf` unterscheiden!