

1 Testataufgaben

1.1 Projekt (Woche 2)

(7 Punkte)

Nun geht es an die Implementierung des Projektes. Da Sie insgesamt zwei Wochen Zeit hierfür haben, sollten Sie in dieser Woche etwa die Hälfte Ihres Programms fertigstellen.

1. Beginnen Sie mit der Implementierung der Funktionen, für die Sie laut Planung aus Übungsblatt 08 verantwortlich sind. Sofern noch keine größeren zusammenhängenden Teile des Projektes lauffähig sind, schreiben Sie kurze Hauptprogramme, um Teilfunktionalitäten testen und demonstrieren zu können.
2. Geben Sie Ihrem Tutor einen mündlichen Zwischenbericht über den Status Ihres Projektes. Überdenken Sie, wie viel Sie schon implementiert haben und was im Rahmen des nächsten Blattes noch zu implementieren ist.
3. Erklären Sie Ihren Gruppenmitgliedern die Stellen in Ihren Funktionen, die strukturierte Datentypen, Dateiverarbeitung und dynamische Speicherverwaltung verwenden. Jedes Gruppenmitglied soll diese Abschnitte dem Tutor erklären können.

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 (optional)

Nach der Wiederholung der Vorlesungsinhalte und der Übungsblätter 01 bis 09 sollte Ihnen die Beantwortung nachfolgender Fragen leicht fallen. Prüfen Sie Ihr Wissen und Verständnis auf eventuelle Lücken und besprechen Sie ggf. mit Ihrem Tutor:

- 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 (Rückgabetypp, 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 = "SSttrriinnngg"`.

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“.