

*Prof. Markus Gross, Bruno Heidelberger, Richard Keiser, Nicky Kern, Edouard Lamboray, Christoph Niederberger, Tim Weyrich, Felix Eberhard, Manuel Graber, Nathalie Kellenberger, Marcel Kessler, Lior Wehrli*

## Uebung 4- Arrays, Strings & Structs

Ausgabe: 24. November 2003  
Abgabe: 1. Dezember 2003  
Autor: Bruno Heidelberger

### 1. Arrays

**2 Punkte**

Welche der folgenden Deklarationsmöglichkeiten sind erlaubt und welche nicht? Bezeichne den Fehler und beschreibe, wie er behoben werden kann. Ansonsten erläutere die Variable und ihre Eigenschaften:

- |                                                |                                                                       |
|------------------------------------------------|-----------------------------------------------------------------------|
| a) <code>char str = "a";</code>                | d) <code>float value[ ];</code>                                       |
| b) <code>short array[4] = { 1, 2 };</code>     | e) <code>unsigned int buffer[3] = { 13, 707, 0, 3000 };</code>        |
| c) <code>char hello[ ] = "hello world";</code> | f) <code>double table[ ][ ] = { { 1.0, 0.0 }, { -1.0, 0.5 } };</code> |

### 2. Strings

**4 Punkte**

Schreibe ein Programm, welches aus dem Vor- und Nachnamen des Benutzers den vollständigen Namen erzeugt. Unabhängig davon, wie sie ursprünglich eingegeben wurden, sollen beide Namensteile im Resultat mit einem Grossbuchstaben beginnen.

- a) Definiere genügend grosse Zeichenketten für den Vor- und Nachnamen, sowie für den zu erzeugenden vollständigen Namen. Frage nun den Benutzer nach Vor- und Nachname.
- b) Erzwinge nun, dass das erste Zeichen von Vor- und Nachname ein Grossbuchstabe ist. Verwende dazu die Funktion `toupper(c)`, welche für einen gegebenen Kleinbuchstaben `c` den entsprechenden Grossbuchstaben zurückgibt. Andere Zeichen werden dabei unverändert zurückgegeben.
- c) Erzeuge den vollständigen Namen, indem du Vor- und Nachname innerhalb der dritten Zeichenkette aneinanderhängst, getrennt durch ein Leerzeichen. Benütze dabei die Funktion `strcat(str1, str2)`, welche die Zeichenkette `str2` an das Ende der Zeichenkette `str1` anhängt. Beachte, dass die Grösse von `str1` für beide Zeichenketten zusammen hinreichend dimensioniert sein muss.
- d) Gib den erzeugten Namen aus, sowie auch dessen Länge. Verwende die Funktion `strlen(str)`, um die Länge einer Zeichenkette `str` zu erhalten.

Hinweis: Um `toupper()` verwenden zu können, muss `<ctype.h>` eingebunden werden. Zusätzlich wird `<string>` für die Funktion `strcat()` benötigt.

### 3. Structs

4 Punkte

In dieser Aufgabe soll die Seitenlänge eines Quadrates berechnet werden, welches den gleichen Inhalt besitzt, wie die zusammengenommene Fläche eines vom Benutzer spezifizierten Kreises und Rechtecks.

a) Definiere zuerst geeignete Structs für die folgenden geometrischen Objekte mit den angegebenen Elementen:

- Kreis mit Radius und Fläche
- Rechteck mit Länge, Breite und Fläche
- Quadrat mit Länge und Fläche

Erzeuge je eine einzelne Variable von den oben definierten Structs. Es sollen im folgenden ausschliesslich diese drei Variablen im Programm verwendet werden.

b) Frage den Radius des Kreises, wie auch die Länge und Breite des Rechtecks vom Benutzer ab. Berechne dann die jeweilige Fläche. (*Achtung!* Wie gesagt, keine zusätzlichen Variablen!).

c) Berechne nun die Fläche und die Seitenlänge des Quadrats. Die dazu hilfreiche Funktion `sqrt ( )` kennst du ja bereits.

d) Gib zum Schluss die Elemente aller drei verwendeten Variablen aus.

### 4. Unix-Einführung: User und Gruppen

(fakultativ)

Wie ihr schon wisst, hat unter Unix jeder Benutzer seinen eigenen Login-Namen. Alle diese Benutzer werden in Gruppen eingeteilt. Für euch lautet die Gruppe `stud`. Dies seht ihr, wenn ihr ein `ls -l` macht und die dritte und vierte Spalte anschaut. Dort seht ihr, dass jede Datei einen Besitzer hat, welcher einer Gruppe zugeordnet ist.

Falls ihr euren Usernamen nicht wisst, so könnt ihr mit dem Kommando `whoami` diesen ausgeben lassen. Dies ist besonders nützlich, wenn ihr verschiedene Logins habt (z.B. Administrator, User, Webmaster, etc.) und oft zwischen diesen wechseln müsst.

Da Unix von Grund auf ein Mehrbenutzersystem ist, kommt es nicht selten vor, dass der Rechner, an dem ihr sitzt, gleichzeitig noch von anderen benutzt wird, welche sich "remote"<sup>1</sup> eingeloggt haben. Wenn ihr den Befehl `who` eintippt, erfahrt ihr, welche User gerade auf der Maschine arbeiten, auf der ihr eingeloggt seid. Ihr erfahrt auch, von wo sie sich eingeloggt haben.

Der Befehl `finger` zeigt euch auf den ersten Blick in etwa dieselbe Information an, kann aber noch mehr. Er zeigt z.B. nicht nur den (meist unverständlichen) Login-Namen an, sondern auch der echte Name des Benutzers. Wenn ihr `finger username` eingibt, erfahrt ihr einige interessante Informationen über diesen Benutzer (z.B. wann der User zum letzten Mal seine E-Mails gelesen hat).

---

1. remote: aus der Ferne, von fern