



Universität Augsburg  
Institut für Informatik

Übung zur Vorlesung Informatik 1

WS 2017/18

Fakultät für Angewandte Informatik

Lehrprofessur für Informatik

PROF. DR. LORENZ, MARIUS BRENDLE, JOHANNES METZGER, LEV SOROKIN

29.11.2017

---

## Übungsblatt 6

---

**Abgabe: 6.12.2017, 12:00 Uhr** (Postkasten der Veranstaltung und E-Mail an Tutor)

- Dieses Übungsblatt muss im Team abgegeben werden (Einzelabgaben sind nicht erlaubt).
- Bitte zur Angabe von Namen, Übungsgruppe und Teamnummer das **Deckblatt** verwenden!
- Die **Zeitangaben** geben zur Orientierung an, wie viel Zeit für eine Aufgabe später in der Klausur vorgesehen wäre; gehen Sie davon aus, dass Sie zum jetzigen Zeitpunkt wesentlich länger brauchen und die angegebene Zeit erst nach ausreichender Übung erreichen.

\* leichte Aufgabe / \*\* mittelschwere Aufgabe / \*\*\* schwere Aufgabe

---

## Aufgabe 21 \* (*Wissensfragen, jede Frage 1 Minute*)

Beantworten Sie möglichst knapp und genau in einem Satz die folgenden Fragen.

### a) (Anweisungen in C-Programmen)

1. Wie lautet die Anweisung, um eine statische Variable **a** vom Typ **double** mit Wert 1.0 anzulegen?
2. Wie lautet die Anweisung, um ein **double**-Feld **v** mit **N** Komponenten unter der Verwendung einer symbolischen Konstante **N** anzulegen?
3. Wie lautet die Anweisung um ein Zeichen einzulesen und in einer Zeichenkette **s** das zweite Zeichen auf das eingelesene Zeichen zu setzen?
4. Wie lautet die Anweisungssequenz, um alle Komponenten eines **int**-Feldes **a** der Länge 10 mit dem Wert 0 zu initialisieren?
5. Geben Sie eine Bedingung an, um zu testen, ob eine Zeichenkette **s** den Wert **"abc"** hat.
6. Geben Sie eine Bedingung an, um zu testen, ob der Inhalt einer Zeichenkette **s** aus weniger als 10 Zeichen besteht.
7. Sei **s** eine Zeichenkette. Geben Sie eine Anweisung an, um in **s** den Wert **"abc"** zu speichern.
8. Geben Sie eine Anweisung an, um an den Inhalt einer Zeichenkette **t** höchstens 2 Zeichen des Inhalts einer Zeichenkette **s** anzuhängen.
9. Was ist der Wert des Ausdrucks `atoi("1.5") - 1`?
10. Was ist der Wert des Ausdrucks `sprintf(s, "%i + %i = %i", 1, 12, 13)` für eine Zeichenkette **s**?

### b) (Speicherverwaltung von Variablen)

1. Was ist die Kommandozeilen-Ausgabe des folgenden Programms?

```
#include <stdio.h>
int main(void)
{
    int a = 10;
    printf("%i", --a);
    return 0;
}
```

2. Was ist die Kommandozeilen-Ausgabe des folgenden Programms?

```
#include <stdio.h>
int main(void)
{
    int a = 10;
    printf("%i", a--);
    return 0;
}
```

3. Was ist die Kommandozeilen-Ausgabe des folgenden Programms?

```
#include <stdio.h>

int orakel(int n) {
    return --n;
}
```

---

```
int main(void)
{
    int a = 10;
    printf("%i", orakel(a));
    return 0;
}
```

4. Was ist die Kommandozeilen-Ausgabe des folgenden Programms?

```
#include <stdio.h>

int orakel(int n) {
    return --n;
}

int main(void)
{
    int a = 10;
    orakel(a);
    printf("%i", a);
    return 0;
}
```

5. Was ist die Kommandozeilen-Ausgabe des folgenden Programms?

```
#include <stdio.h>

int g = 10;

int orakel(void) {
    return --g;
}

int main(void)
{
    orakel();
    printf("%i", --g);
    return 0;
}
```

6. Was ist die Kommandozeilen-Ausgabe des folgenden Programms?

```
#include <stdio.h>

int g = 10;

int orakel(int n) {
    return --n;
}

int main(void)
{
    orakel(g);
    printf("%i", g);
    return 0;
}
```

7. Was ist die Kommandozeilen-Ausgabe des folgenden Programms?

---

```
#include <stdio.h>

int orakel() {
    static int s = 10;
    return --s;
}

int main(void)
{
    orakel();
    printf("%i", orakel());
    return 0;
}
```

---

## Aufgabe 22 \*\* (*Funktionen für Zeichenketten, 20 Minuten*)

In jeder Teilaufgabe sollen Sie eine Funktion zur Verwaltung von Zeichenketten implementieren. Teilweise handelt es sich dabei um Nachimplementierungen von Bibliotheksfunktionen.

Es wird jeweils kein Hauptprogramm zum Testen Ihrer Implementierung verlangt, Sie sollten sich aber jeweils eines schreiben, damit Sie sicher sind, dass Ihre Abgabe auch stimmt. Außerdem ist das eine gute Übung und führt dazu, dass Sie lernen, korrekte Programme ohne viel Nachzudenken schreiben zu können.

a) (\*\*, 4 Minuten)

Implementieren Sie die Bibliotheksfunktion `strcat` aus der Header-Datei `string.h` nach, ohne dazu andere Bibliotheksfunktionen außer `strlen` zu benutzen. Ihre Funktion soll den Namen `my_strcat` haben.

b) (\*\*\*, 6 Minuten)

Implementieren Sie die Bibliotheksfunktion `strncpy` aus der Header-Datei `string.h` nach, ohne dazu irgendwelche Bibliotheksfunktionen zu benutzen. Ihre Funktion soll den Namen `my_strncpy` haben.

c) (\*\*, 6 Minuten)

Implementieren Sie eine Funktion `unsigned int strcount(const char * cs, const char * ct)`, die die Anzahl der Zeichen in `cs`, die in `ct` vorkommen, liefert.

Beispiel: Der Rückgabewert von `strcount("HALLO", "LAGER")` ist 3.

d) (\*\*, 4 Minuten)

Implementieren Sie eine Funktion `int isprefix(const char * cs, const char * ct)`, die einen von 0 verschiedenen Wert liefert, falls `cs` mit `ct` beginnt, und andernfalls 0 liefert.

Beispiel: Der Rückgabewert von `isprefix("Informatik", "Info")` ist von 0 verschieden.

---

### Aufgabe 23 \*\* (*Zufallszahlen, 12 Minuten*)

In den folgenden Aufgaben soll jeweils eine Funktion zur Generierung von Zufallszahlen in einem bestimmten Zahlenbereich implementiert werden. Benutzen Sie dazu jeweils geeignet die Bibliotheksfunktion `rand`, s.d. der Startwert jeweils mit `srand` neu gesetzt werden kann. Einen geeigneten Prototyp entwerfen Sie jeweils dazu selbst.

Es wird jeweils kein Hauptprogramm zum Testen Ihrer Implementierung verlangt, Sie sollten sich aber jeweils eines schreiben, damit Sie sicher sind, dass Ihre Abgabe auch stimmt. Außerdem ist das eine gute Übung und führt dazu, dass Sie lernen, korrekte Programme ohne viel Nachzudenken schreiben zu können.

a) (\*, Lotto, 2 Minuten)

Implementieren Sie eine Funktion, die zufällig eine ganze Zahl zwischen 1 und 49 (jeweils einschließlich) zurückgibt. Wählen Sie dazu insbesondere selbst einen geeigneten Prototyp für diese Funktion mit möglichst spezifischen Datentypen.

b) (\*\*, Farbauswahl, 4 Minuten)

Implementieren Sie eine Funktion, die zufällig eine der Zeichenketten `"rot"`, `"blau"` oder `"gelb"` zurückgibt.

c) (\*\*, Zufälliges Zeichen, 6 Minuten)

Implementieren Sie eine Funktion, die ein zufälliges ASCII-Zeichen aus der Menge der Klein- und Großbuchstaben, Ziffern und der Teilmenge der Sonderzeichen mit ASCII-Codes zwischen 33 bis 47 zurückgibt.

---

### Aufgabe 24 \*\*\* (*Komplexe Benutzereingaben, 12 Minuten*)

In dieser Aufgabe sollen Sie ein compilierbares Hauptprogramm implementieren, das folgende Benutzereingabe erwartet, einliest, auf Gültigkeit überprüft und in einer geeigneten Variablen speichert, und schließlich auf Kommandozeile ausgibt, ob die Eingabe gültig war oder nicht:

Die Benutzereingabe soll aus einer Ziffernfolge mit maximal 10 Ziffern bestehen. Beginnt die Ziffernfolge mit der Ziffer 0, so darf danach kein weiteres Zeichen mehr kommen.

Das Einlesen soll dabei in einer separaten Funktion `int read_int(char input[])` erfolgen, die gültige Eingaben in `input` ablegt und 1 zurückgibt, und bei ungültigen Eingaben den Eingabepuffer leert den Wert 0 zurückgibt.

Die Grenze der gültigen Eingabelängen verwalten Sie durch eine geeignete symbolische Konstante.

Sie müssen keine Pufferfehler berücksichtigen.