

# Informatik 1 für Regenerative Energien

## Klausur vom 5. Februar 2014

Jörn Loviscach

Versionsstand: 4. Februar 2014, 23:43



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

*Drei Punkte pro Aufgabe. Mindestpunktzahl zum Bestehen: 15 Punkte. Hilfsmittel: maximal vier einseitig oder zwei beidseitig beschriftete DIN-A4-Spickzettel beliebigen Inhalts, möglichst selbst verfasst oder zusammengestellt; kein Skript, keine anderen Texte, kein Taschenrechner, kein Computer, kein Handy und Ähnliches.*

Name	Vorname	Matrikelnummer	E-Mail-Adresse, falls nicht in ILIAS

### Fingerübungen

1. Welches Bitmuster (acht Bit mit Zweierkomplement) steht für die Zahl  $-17$ ? Welches für die Zahl  $7$ ? Wie rechnet man dann  $-17 + 7$  aus? Schreiben Sie das mit Bits hin.
2. Gegeben sind die C-Variablen `unsigned char a = 0x8A;` und `unsigned char b = 0x9F;`. Geben Sie in Hexadezimal an, welche Werte die drei C-Ausdrücke `a|b` und `a&b` und `a^b` bei Rechnung mit acht Bit haben.
3. Geben Sie durch Klammern an, wie die Bedingung im `if` des folgenden C-Fragments ausgewertet wird. Was sind jeweils die Teilergebnisse?

```
double a = 1.0;
bool b = true;
int c = 7;
if(a + 2.0 * 3.0 < 10.0 && ! b || c % 4 == 3)
{
    // ...
}
```

4. Die folgende C-Funktion soll die übergebene Zahl im Zahlensystem zur Basis 7 darstellen. Dazu wird der Funktion eine nullterminierte Zeichenkette übergeben, in die das Ergebnis geschrieben werden soll (soweit Platz ist). Es sind aber drei Fehler in der Funktion. Finden und korrigieren Sie diese.

```
void inSiebenersystem(unsigned int z, char s)
{
    int n = strlen(s) - 1;
    while(n >= 0 || z > 0)
    {
        s[n] = '0' + z%7;
        n--;
        z *= 7;
    }
}
```

5. Gegeben ist diese Datenstruktur für ein Vereinsmitglied:

```
struct Mitglied
{
    char vorname[16];
    char nachname[16];
    int geburtsjahr; // mit vier Stellen: 1980 usw.
};
typedef struct Mitglied Mitglied;
```

Schreiben Sie eine C-Funktion, die ein Array von Mitgliedern sowie deren Anzahl entgegen nimmt und als Ergebnis zurück gibt, wie viele der vor 1990 Geborenen einen Nachnamen haben, der mit dem Buchstaben L beginnt.

6. Schreiben Sie eine C-Funktion

`bool vergleicheUmgekehrt(char a[], char b[])`, die zwei Zeichenketten (nullterminiertes C-Strings) entgegen nimmt und zurück meldet, ob die eine Zeichenkette die umgedrehte Version der anderen Zeichenkette ist, wie zum Beispiel bei `hallo` und `ollah`. **Achtung:** Die Länge des Texts wird nicht gesondert übergeben. Die Funktion `strlen` ist erlaubt. Lesen Sie nicht jenseits der Enden der Arrays mit den Zeichenketten.

### Kreative Anwendung

7. Zeichnen Sie ein Flussdiagramm für dieses Stück C-Programmcode. Beachten Sie, dass sich im Flussdiagramm nicht direkt `while` und `for` verwenden lassen.

```
int a = 3;
int b = 4;
while(b < 100)
{
    b += a;
}
for(int i = 50; i > a; i--)
{
    b = b/2 + i;
}
```

8. Ersetzen Sie das `switch` in diesem Stück C-Code durch eine Konstruktion mit `if` und `else`, die für alle Werte von `a` und `u` das jeweils gleiche Ergebnis liefert wie das `switch`:

```
enum Ampel {rot, gelb, gruen};
typedef enum Ampel Ampel;

Ampel a = rot;
int u = 1;

switch(a)
{
case rot:
    u++;
case gelb:
    u += 2;
case gruen:
    u += 3;
    break;
default:
    u = 13;
}
```

9. Ein Algorithmus erhält ein *aufsteigend sortiertes* Array von  $n$  Zahlen. Was ist die Zeitkomplexität (worst case) für folgende Aufgaben:
- die Summe der  $n$  Zahlen bestimmen,
  - die kleinste der  $n$  Zahlen finden,
  - feststellen, ob eine der  $n$  Zahlen durch eine andere dieser Zahlen teilbar ist.

Geben Sie jeweils die kleinste dieser Mengen an:  $O(1)$ ,  $O(\log n)$ ,  $O(n)$ ,  $O(n \log n)$ ,  $O(n^2)$ ,  $O(n^3)$ .

10. Welche Werte stehen nach Ausführung dieser Zeilen C-Code in den Variablen  $a$ ,  $b$ ,  $c$ ?

```
int u[3] = {13, 42};
int* v = u + 1;
int w = v[0];
int* x = &w;
(*x)--;
v++;
int a = *v;
int b = w;
int c = x[0];
```

11. Welche Zahlenwerte (dezimal angegeben, mit Vorzeichen) stehen nach Ausführung des folgenden C-Codes auf einem 16-Bit-Rechner in den Variablen  $c$  und  $d$ ? Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.

```
int a = 100 * 256;
int b = 28 * 256;
int c = (a + b)/2;
int d = a/2 + b/2;
```

12. Schreiben Sie diese C-Funktion so um, dass man sofort sehen kann, was sie tut. (Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.)

```
int g(int a)
{
    int b = a;
    for(int i = 0; i < a; i++)
    {
        b--;
    }
    return b;
}
```