

# Informatik 1 für Regenerative Energien

## Klausur vom 18. September 2013

Jörn Loviscach

Versionsstand: 18. September 2013, 08:23



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

1. Welches Bitmuster (acht Bit mit Zweierkomplement) steht für die Zahl  $-11$ ? Welches für die Zahl  $6$ ? Wie rechnet man dann  $-11 + 6$  aus? Schreiben Sie das mit Bits hin.
2. Gegeben sind die C-Variablen `unsigned char a = 0xE1;` und `unsigned char b = 0x3C;`. 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?

```
int a = 5;
bool b = false;
if(a >= a % 3 / 4 + 3 || ! b && a == 1)
{
    // ...
}
```

4. Die folgende C-Funktion soll die Quersumme (d. h. die Summe der Dezimalziffern) bestimmen. Beispiele: Die Zahl 123 hat die Quersumme  $1 + 2 + 3 = 6$ . Die Zahl 2009 hat die Quersumme  $2 + 0 + 0 + 9 = 11$ . Es sind aber drei Fehler in der Funktion. Finden und korrigieren Sie diese.

```
unsigned int bildeQuersumme(unsigned int x) // Hat drei Fehler!
{
    unsigned int q;
    while(x < 0)
    {
        q += x / 10;
        x /= 10;
    }
    return q;
}
```

5. Gegeben ist diese Datenstruktur:

```
struct Person
{
    char name[16];
    int alter; // in Jahren
};
typedef struct Person Person;
```

Schreiben Sie eine C-Funktion

```
int zaehleVolljaehrige(Person personen[],
                      int anzahl, char name[])
```

die als Ergebnis zurückgibt, wie viele Personen des Arrays `personen` den Namen `name` haben und 18 Jahre alt oder älter sind. In `anzahl` wird der Funktion die Anzahl der Einträge des Arrays `personen` übergeben. Sie können `strcmp` aus `string.h` verwenden.

6. Schreiben Sie dieses C-Programmfragment mit `switch` statt mit `if`:

```
int a;
int b;
// ...
if(a == 10)
{
    b = a + 3;
}
else if(a > 4 && a <= 7)
{
    b = a + 4;
}
else
{
    b = 5;
}
```

7. Ein Programm soll so lange Zahlen vom Benutzer abfragen, bis der eine 0 eingibt. Danach soll das Programm die größte und die kleinste eingegebene Zahl ausgeben (die 0 am Ende dabei nicht einbezogen). Es wird niemals als Erstes die 0 eingegeben. Lösen Sie das mit einem Flussdiagramm. Dieses darf Eingabe, Ausgabe, Zuweisung, Addition, Multiplikation, Vergleich/Verzweigung benutzen, aber keine Arrays.
8. Schreiben Sie eine C-Funktion `int liesBinaer(char b[])`, die eine Zeichenkette (einen nullterminierten C-String) entgegen nimmt, diese beginnend von links so weit wie möglich als Binärzahl liest und den Wert als `int` zurückliefert. Beispiel: `110` wird zu `6`, `10abc` wird zu `2`, `abc` wird zu `0`, `abc110` wird auch zu `0`. Die Funktion `strlen` ist erlaubt. Lesen Sie nicht jenseits des Endes des Arrays mit der Zeichenkette.
9. Ein Algorithmus habe bei einer Eingabegröße von  $n$  die Laufzeit  $\frac{n^5}{n^2+7n+3}$ . Ist das  $O(n)$ ? Ist das  $O(n^2)$ ? Ist das  $O(n^3)$ ?
10. Welche Werte stehen nach Ausführung dieser Zeilen C-Code in den Variablen `a`, `b`, `c`?

```
int x[6] = {1, 2, 3};
int y = 4;
int *z = &y;
y++;
int a = *z;
z = x + 1;
int b = z[1];
int c = x[5];
```

11. Welcher Wert steht nach Ausführung des folgenden C-Code auf einem 16-Bit-Rechner in der Variablen  $y$ ? Welcher in der Variablen  $z$ ? (Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.)

```
unsigned int x = 16;
unsigned int y = (x + 240) * 256;
unsigned int z = x + 0xFFFF;
```

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 f(int u)
{
    u--;
    int y = 13;
    for(int i = 0; i <= y; i++)
    {
        u--;
    }
    return u;
}
```