

# Informatik 1 für Regenerative Energien

## Klausur vom 21. September 2011

Jörn Loviscach

Versionsstand: 27. September 2011, 18:50



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

### Fingerübungen

1. Welches Bitmuster (acht Bit mit Zweierkomplement) steht für die Zahl  $-11$ ? Welches für die Zahl  $14$ ? Wie rechnet man dann  $(-11) + 14$  aus? Schreiben Sie das mit Bits hin.
2. Gegeben sind die C-Variablen `unsigned char a = 0x3A;` und `unsigned char b = 0x76;`. Geben Sie in Hexadezimal an, welche Werte die drei C-Ausdrücke `~a` 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?

```
bool a = false;
int b = 9;
if(a || b % 3 + 5.0f > 1.0f && ! a)
{
    // ...
}
```

4. Die folgende C-Funktion soll feststellen, ob die übergebene Zahl (nur Zahlen ab 2 aufwärts sind erlaubt) eine Primzahl ist, also eine Zahl, die nur durch 1 und durch sich selbst ohne Rest teilbar ist. Beispiele: Für 13 soll sie `true`

zurückliefern; für 42 soll sie `false` zurückliefern. Es sind aber drei Fehler in der Funktion. Finden und korrigieren Sie diese.

```
void isPrime(unsigned int x) // Hat drei Fehler!
{
    for(unsigned int i == 2; i < x; i++)
    {
        if(x%i == 1) // Teilbarkeit prüfen
        {
            return false;
        }
    }
    return true;
}
```

5. Gegeben ist die Datenstruktur `Item` für einen Buchungsposten:

```
struct Item
{
    char name[8]; // Bezeichnung
    int amount; // Betrag
};
typedef struct Item Item;
```

**Schreiben Sie eine C-Funktion**

`int addAmounts(Item items[], int number, char initial)`, die als Ergebnis die Summe der Beträge aus allen Buchungsposten im Array `items` zurückgibt, deren Namen<sup>c1</sup> mit dem Buchstaben `initial` anfangen. In `number` wird der Funktion die Anzahl der Buchungsposten des Arrays `items` übergeben.

c1 text added by jl

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

```
int x = 42;
int y = 0;
// ...
switch(x)
{
    case 1:
    case 2:
        y = x;
    case 3:
        y += 10;
        break;
    default:
        y = 11;
}
```

### Kreative Anwendung

7. Ein Programm soll eine ganze Zahl vom Benutzer abfragen und dann feststellen, ob diese Zahl eine Zweierpotenz (1, 2, 4, 8, ...) ist. Es soll entsprechend „Ja“ oder „Nein“ ausgeben. Lösen Sie das mit einem Struktogramm. Dieses darf Eingabe, Ausgabe, Zuweisung, Vergleich, Addition, Multiplikation, Verzweigung, Schleife benutzen. (Ignorieren Sie etwaige Probleme mit Überlauf oder Rundung.)
8. Schreiben Sie eine C-Funktion `bool findAB(char a[])`, die einen Text (nullterminierten C-String) entgegen nimmt und zurück meldet, ob darin die Zeichenfolge AB enthalten ist. Achtung: Die Länge des Texts wird nicht gesondert übergeben. Die Funktion `strlen` ist erlaubt. Lesen Sie nicht jenseits des Endes des Arrays mit der Zeichenkette.
9. Die folgende Funktion wird mittels `f(3)` aufgerufen. Geben Sie an, welcher Wert danach in der Variablen `a` steht. (Geben Sie möglichst auch Zwischenschritte an, damit Ihr Gedankengang nachvollziehbar ist.)

```
int f(unsigned int x)
{
    static int a = 0;
    a++;
    if(x == 0)
    {
        return 1;
    }
    else if(x == 1)
    {
        return 1;
    }
    else return f(x-1)+ f(x-2);
}
```

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

```
int z[] = {13, 42, 100};
int a = *(z + 1);
int* y = z + 1;
int b = y[1];
(*y)--;
int c = z[1];
```

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

```
unsigned int x = 16;
unsigned int y = x*4*32*32;
```

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 x)
{
    bool a = false;
    if(x < 0)
    {
        x++;
        a = true;
    }
    else
    {
        x--;
    }
    if(a)
    {
        return x-1;
    }
    return x+1;
}
```