

# Aufgaben zu Basiswissen IT

## 1. Der (nicht so) geheime Code

Was mag dies bedeuten:

```
01010011 01111001 01101101 01100010 01101111 01101100 01100101
00100000 01110101 01101110 01100100 00100000 01010011 01101001
01100111 01101110 01100001 01101100 01100101
```

Hinweis: ASCII-Code

Wie lautet der String in Hexadezimalzahlen, in Dezimalzahlen, in Oktalzahlen?

## 2. Na logisch logisch

In der (binären) Logik benutzt man oft sogenannte Wahrheitstafeln. Die elementaren logischen Operation lassen sich damit übersichtlich darstellen. Schreibt man 1 für wahr und 0 für falsch, dann sehen die Wahrheitstafeln für NOT , OR und AND so aus:

	1	0
NOT	0	1

OR	1	0
1	1	1
0	1	0

AND	1	0
1	1	0
0	0	0

- (a) Drücken Sie folgende Wahrheitstafel mit Hilfe der elementaren Operationen NOT, OR, AND aus.

XOR	1	0
1	0	1
0	1	0

- (b) Beweisen Sie (mit Hilfe von Wahrheitstafeln) die *de Morganschen* Gesetze (NOT hat Vorrang vor AND/OR) :
- a  $a \text{ AND } b = \text{NOT}(\text{NOT } a \text{ OR NOT } b)$
  - a  $a \text{ OR } b = \text{NOT}(\text{NOT } a \text{ AND NOT } b)$
- (Hier zeigt sich, dass man für die binäre Logik im Prinzip nur zwei Operationen braucht.)
- (c) Wieviel verschiedene Wahrheitstafeln gibt es?

## 3. Little Big Endian

Eine lange Ganzzahl (`long`, `int`) belegt normalerweise 4 Bytes im Speicher. Es gibt zwei Konventionen dies zu tun: Little Endian und Big Endian. Bei Big Endian ist das höchstwertige Byte (Most Significant Byte) an der Adresse `ADR` gespeichert, das nächste Byte an der Adressen

ADR+1 usw., bei Little Endian ist es genau umgekehrt. Intel Rechner gehorchen der Little Endian Konvention. Eine IP-Adresse besteht auch aus 4 Byte, aber im Big Endian-Format, laut Protokollspezifikation. Auf Intel-Rechner besteht also das Problem, die Byte Reihenfolge umzukehren. Nehmen wir an, wir hätten einen primitiven Prozessor, der ein internes Register (Arbeitsspeicher) besitzt und einen Stapelspeicher, den sogenannten Stack. Ein Stapelspeicher, auch Kellerspeicher genannt, funktioniert nach dem LIFO-Prinzip (Last-In-First-Out), d.h. der letzte Wert der auf den Stapel gelegt wurde (PUSH), kommt als erstes wieder heraus (POP). Desweiteren kann der Prozessor Werte aus dem Hauptspeicher in sein Register laden und wieder zurückspeichern. Der Prozessor versteht also folgende Befehle:

Befehl	Operation
MOV <i>adr</i>	Holt das Byte von Adressen <i>adr</i> in das Register
STO <i>adr</i>	Speichert den Inhalt des Registers bei Adresse <i>adr</i>
PUSH	Schiebt den Inhalt des Registers auf den Stack
POP	Holt den obersten Wert des Stacks ins Register

Schreiben Sie ein kurzes Programm, das die Reihenfolge der Bytes an den Speicherstellen 0x01, 0x02, 0x03, 0x04 umkehrt.