EI Info 2010-11	*INFORMATIK* 1. Probeklausur	

1. Aufgabe
Was ist ein Bit? Was ist ein Byte?

Ein Bit repräsentiert in der Informatik meistens 0 oder 1 (was einem Wahrheitswert entspricht oder wahr/falsch, an/aus usw.) und ist damit die kleinste Informationseinheit im Rechner. 8 Bit werden zu einem Byte zusammengefasst.

2. Aufgabe
Wie wird die Zahl 100 (Zehnerstellensystem!) im Binärsystem geschrieben? Wie im Hexadezimalsystem? Welches ist die größte darstellbare Zahl des Binärsystems, wenn man ein Byte zur Verfügung hat?

100=64+32+4, also als Byte: 0110 0100. Im Hexsystem: 64. Dazu kommt man am besten über 0110=6 bzw. 0100=4. Mit der hinteren Frage war nach der Anzahl der Zustände gefragt. Maximal 256.

3. Aufgabe
Wie addiert der PC die Binärzahlen 1000 0001 und 0100 1001? Wie lautet das Ergebnis? Überprüfe im Zehnersystem. Kann es Probleme mit der Addition zweier Bytezahlen geben, wenn im Speicher für das Ergebnis ebenfalls ein Byte zur Verfügung steht?

Zahlen: Übertrag: Endergebnis:
1000 0001
0100 1001 0000 0000 0000 0000

jetzt wird addiert, also 1+1,0+0 usw:

1000 0001
0100 1001 0000 0010 1100 1000

jetzt testet der PC, ob der Übertrag leer ist. Ist er nicht. Daher überschreibt er die Anfangszahlen wie folgt:

0000 0010
1100 1000 0000 0010 1100 1000

Jetzt werden Übertrag und Ergebnis wieder auf Null gesetzt:

0000 0010
1100 1000 0000 0000 0000 0000

Nun wird wieder addiert:

0000 0010
1100 1000 0000 0000 1100 1010

Dieses Mal gab es keinen Übertrag! Der Rechner prüft das, ist zufrieden und gibt das Ergebnis 1100 1010 aus. Notfalls hätte er die ganze Prozedur wiederholt…

Im Extremfall kann es zu einem Speicherüberlauf kommen. Beispiel:
1000 0000 + 1111 0000 lässt sich nicht mehr in ein einzelnes Byte schreiben.

4. Aufgabe
Wie kodiert die Programmiersprache html Farben? Male das Schwarzweißbild (schwarz=0, weiß=1), welches ein 4x4-Bild ist, das mit dieser Zeichenfolge codiert wurde: 1001011010011111.

Nach der RGB-Regel (also physikalisch). Meint: Die Farbe hat einen Rot-, einen Grün- und einen Blauanteil. Die werden dann gemischt. für jede dieser drei Farben sind 256 verschiedene Abstufungen möglich. Die Bildschirme bestehen normalerweise je Pixel aus drei „Lichtpunkten", einem roten, einem grünen und einem blauen, die dann je nachdem verschieden hell leuchten…

1001011010011111 meint damit offensichtlich

1001
0110
1001
1111

und das sieht dann so aus:

xoox
oxxo
xoox
xxxx (ein o meint einen schwarzen Punkt, x ist weiß)

5. Aufgabe
Wie viele Bytes kann deine 1-Terabyte-Festplatte speichern?! Nimm an, ein durchschnittliches Buch enthält 500 Seiten zu je 100 Zeilen mit jeweils 100 Zeichen. Wieviele solcher Bücher könntest du auf deiner Festplatte speichern? Schätze so ab, ob man alle Bücher, die bisher geschrieben wurden, auf deiner Platte speichern könnte. Warum verbrauchen Filme mehr Speicherplatz als Texte?

Tera = 1000 Giga = 1000 (1000 Mega) = 1000 (1000[Millionen]) und damit entspricht ein Terabyte 10^{12} bzw. eine Millionen Millionen Byte.

Ein 500-Seitenbuch mit 100x100 Zeichen je Seite besteht aus 500x10000 oder 5 Mio. Zeichen. 2 Dieser Bücher sind 10 Mio Zeichen. 200 dieser Bücher haben 1000 Mio Zeichen. Wir haben aber eine Millionen einer Millionen Zeichen Platz, also noch einmal einen Faktor 1000. Und so können wir 200.000 Bücher einspeichern. Im deutschen Sprachraum erscheinen im Moment ca. 100.000 Bücher jährlich (wobei es hier sicher viele Dopplungen gibt). Insoweit wird das nicht reichen.

Filme speichern ja neben Bild auch noch Ton ab. Beides nimmt viel mehr Speicher in Anspruch als ein einfaches Zeichen (das verbraucht ja nur 1Bit). Alleine ein Pixel im RGB mit je einem Byte „verschluckt" schon Speicher für 3 Zeichen… Je nach Auflösung wird das „teuer"!

6. Aufgabe
Wieso ist die Datei test.txt viel kleiner als die daten test.doc, auch wenn sie denselben Text enthalten?

Weil bspw. Word zahlreiche Zusatzcodierungen um den Text herum abspeichert (Textformatierungen, Schriftart usw.).

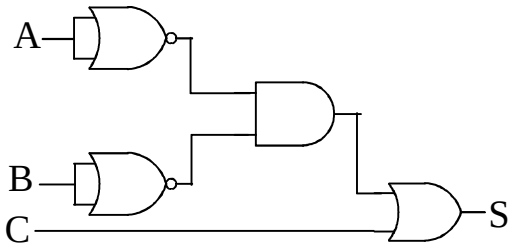
7. Aufgabe
Konstruiere eine Aussage, die immer wahr sein muss. Konstruiere eine Aussage, die immer falsch ist. Bestimme den Wahrheitswert der folgenden Aussage. Dabei ist A falsch, B und C jedoch wahr: $(A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C)$.

Ich schreibe „Java":

A && !A ist immer falsch. A || !A ist übrigens immer wahr.

Zum zweiten Ausdruck: erste Klammer ist wahr, da A=f, B=w, !C=f ist. Die zweite Klammer ist wahr, da A=f, !B=f, aber C=w gilt. Die dritte Klammer ist falsch, denn A=f, !B=!C=f und somit ist der gesamte Ausdruck falsch.

8. Aufgabe
Entwickle für folgendes Schaltnetz mit den Eingängen A, B und C eine Schaltwerttabelle und notiere einen dazu passenden Booleschen Term:



Ehrlich gesagt habe ich jetzt nicht nachgesehen, welches Zeichen welchem logischen Operator entspricht. In der Arbeit schreibe ich das dann rein, damit ihr nichts auswendig lernen müsst.

Ich tue mal so, als wäre das erste Symbol rechts von A (bzw. B) der !-Operator. Die werden mit && Verknüpft und dann mit einem || mit C. Das ergibt S. Also gilt:

S = (!A && !B) || C. Das ist der boolesche Term. Die Schaltwerttabelle meint, was jeweils ist, wenn A, B bzw. C wahr oder falsch (1 oder 0) sind:

Notation: ABC als 111 meint alle sind wahr!

S = 1 für 111, 101, 001, 000
S = 0 für 010, 100, 011, 110

Sowas als Tabelle wäre eine Schaltwerttabelle. Es geht aber auch wie gerade gezeigt.

9. Aufgabe
Notiere für die Funktion „tanze()" einen Pseudocode. Die Funktion soll aus dieser Abfolge bestehen: Linksdrehung, Rechtsdrehung, Linksdrehung, Schritt vor, Schritt zur Seite, Schritt nach hinten, Schritt nach rechts. Verwende dazu nur die Funktionen bewegen() und dreheRechts(). Du kannst dir natürlich aus diesen beiden „Atomen" weitere Hilfsfunktionen schreiben! Wo musst du im Roboterszenario diese Funktion implementieren (: was heißt dieses Wort?), damit die Roboter Robby, Robson und Robita tanzen können?

Pseudocode:

```
funktion tanze()
{
        dreheLinks();
        dreheRechts();
        dreheLinks();
        bewegen();
        bewegenSeite();
        bewegenHinten();
        bewegenRechts();
}
```

Dabei muss ich folgende Hilfsbefehle einführen:

1. **dreheLinks enthält 3x dreheRechts.**
2. **bewegenSeite besteht aus dreheLinks, bewegen, dreheRechts.**
3. **bewegenHinten besteht aus 2x dreheRechts, bewegen und dann wieder 2xdreheLinks**
4. **bewegenRechts besteht aus dreheRechts, bewegen, dreheRechts.**

Geht aber auf 1000 verschiedene Weisen!

10. Aufgabe
Wann hebt Robby den Akku auf? Hier der act-Code:

```
{
        if (akkuAufFeld() && (wandVorne() || (wandLinks() && !wandRechts()) ) )
                {
                        akkuAufnehmen();
                }
        else
                {
                        tanze();
                }
}
```

Zuerst einmal muss ein Akku daliegen. Das alleine reicht aber nicht, denn entweder soll eine Wand vor Robby sein oder es gibt gleichzeitig eine Wand links und keine Wand rechts. In allen anderen Fällen hebt Robby den Akku nicht auf, sondern tanzt.

11. Aufgabe
Robby steht in einem langen geraden Gang (links und rechts zur Laufrichtung sind Wände). Nach einer unbekannten Anzahl von Schritten kommt ein Ausgang. Dabei ist Robby auch nicht

klar, ob dieser oben oder unten kommt. Robby soll diesen Ausgang finden und für Robita an der Stelle, wo es zum Ausgang geht, eine Schraube ablegen. *Zusatz: Danach geht er durch den Ausgang, um mit Robson zu tanze().* Notiere im Pseudocode, wie Robby den Ausgang findet und die Schraube ablegt. *Zusatz: Notiere, wie Robby zusätzlich durch diesen Ausgang geht, Robson findet (der dann in direkter Linie steht) und mit ihm tanzt.*

man schreibt 2 if-Abfragen:

if (!wandRechts()) bzw. if (!wandLinks()). Für den Fall, dass gleichzeitig oben und unten ein Loch kommt, wird halt unten (durch wandRechts) aufgefüllt. Also:

```
if (!wandRechts())
{
        dreheRechts();
        bewegen();
        schraubeAblegen();
}
else
{
        bewegen();
}
```

und das gleiche für das Loch oben:

```
if (!wandLinks())
{
        dreheRechts();
        bewegen();
        schraubeAblegen();
}
else
{
        bewegen();
}
```

Den Zusatz fügt man in die if-Bedingung ein nach dem Schraubeablegen:

```
...
if (robbyVorne())              [hier: einfach Code von wandVorne() „klauen"...]
{
        tanze();
}
else
{
        bewegen();
}
...
```

12. Aufgabe

Robita geht einen Gang entlang. Ab und an liegt ein Akku herum. Diese soll Robita aufnehmen und dafür eine Schraube ablegen. Liegen jedoch zwei Akkus hintereinander, so soll sie nur den vorderen aufheben! Implementiere im Pseudocode.

In Robitas Code folgende if-Bedingung einfügen:

```
if (akkuAufFeld() && !schraubeHinten())
{
        akkuAufnehmen();
        schraubeAblegen();
}
else
{
        bewegen();
}
```

Die schraubeHinten-Methode kann man von wandVorne klauen und entsprechend abändern!