



Unsere erste Informatik-Klausur! Du kannst deinen GTR verwenden. Achte auf eine übersichtliche und gut erläuterte Darstellung! **(Bearbeitungszeit: 60 Minuten)**

1. Aufgabe

(7 Punkte)

- a) Was ist ein Bit? Was ist ein Byte? Was ist ein Terabyte?

Ein Bit ist die kleinste Informationseinheit in der Informatik. Beispiele: Strom an bzw. Strom aus, An – Aus, 0 – 1 usw. Ein Byte entspricht 8 Bits. Tera kommt nach Giga. Giga nach Mega. Mega steht für Millionen, also ist 1 Megabyte = 1.000.000 Byte (wobei eigentlich nicht 1000, sondern 1024 in der Informatik dem Kilo und 1024² dem Mega entspricht). Dann x1000 auf Giga und nochmal x1000 und man ist bei Tera. Das sind insgesamt 10¹² Bytes.

- b) Übersetze folgende Binär-Bytes ins Dezimalsystem und ins Hexadezimalsystem:

0010 0000₂, 1000 0001₂, 1011 1111₂.

1. Zahl: 32, 20₁₆, 2. Zahl: 129, 81₁₆, 3. Zahl: 191, BF₁₆.

- c) Stelle die folgenden Hexadezimalsystem-Zahlen im Dezimalsystem dar:

01₁₆, 99₁₆, F9₁₆.

1. Zahl: 16, 2. Zahl: 153, 3. Zahl: 249.

- d) Stelle die Zehnersystemzahl 119 im Hexadezimalsystem und als Byte im Binärcode dar.

Hex: 77, Byte: 01110111.

- e) Wieso eignet sich die Hexadezimaldarstellung gut, um Binär-Bytes darzustellen?

Die Hexadezimaldarstellung kann genauso wie die Byte-Darstellung 256 verschiedene Zustände beschreiben, umfasst aber nur 2 Zeichen, während ein Byte aus 8 Zeichen besteht. Sie ist damit oft praktischer zu notieren!

2. Aufgabe

(4 Punkte)

- a) Welche Komponenten braucht ein Rechner, damit ein User zwei ganze Zahlen im Zahlenbereich von 0 bis 200 eingeben kann und der Rechner diese intern addieren kann, um anschließend das Ergebnis auszugeben?

Zwei Speicher je 1 Byte (das bis zu 256 Zahlen darstellen kann), dann einen Speicher für das Ergebnis mit 2 Bytes (das Ergebnis kann mit 400 ja über 256 liegen). Ein Zwischenspeicher für die Überträge. Hier reicht im Prinzip wieder 1

Byte, wenn man es geschickt anstellt. Daneben natürlich die Recheneinheit und ein Eingabesystem sowie ein Ausgabesystem.

b) Beschreibe kurz, wie die Datenverarbeitung intern abläuft.

Die Zahlen werden binär addiert. Dabei auftretende Überträge kommen in einen Zwischenspeicher. Die „Additionszahl“ wird wie der Zwischenspeicher als neue 1. Zahl bzw. 2. Zahl gesetzt. Nun wiederholt sich der Vorgang, bis der Zwischenspeicher leer ist. Dann ist die „Additionszahl“ auch das Ergebnis der ursprünglichen Addition.

3. Aufgabe

(3 Punkte)

Vereinfache folgende boolesche Terme so weit wie möglich. Forme dabei übersichtlich um!

- a) $(A \vee \neg A)$
b) $(B \vee C) \wedge (A \vee C) \wedge (C \vee A) \wedge (C \vee \neg B)$

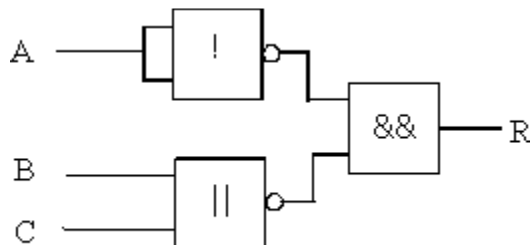
a) ist immer wahr!

Da in b) in der ersten und der letzten Klammer einmal B und einmal nichtB mit C vereinigt ist, muss eine der beiden Klammern falsch sein, außer C ist wahr. Nur dann kann der komplette Ausdruck wahr sein! Also ist „C“ eine vereinfachte Version von Ausdruck b)!

4. Aufgabe

(2 Punkte)

In dem Schaltnetz unten gibt es drei Eingänge A, B und C und einen Ausgang R.



a) Welchen Wert hat R, wenn A=1, B=1 und C=0 gilt?

Nicht A macht 0, das wegen dem UND 0 bleibt. R ist 0.

b) Notiere für R einen passenden Booleschen Term.

R = nicht A UND (B ODER C)

5. Aufgabe

(2 Punkte)

Bestimme nachvollziehbar (!) den Wahrheitswert (w/f) folgender Aussage:

$$((3 < -4) \vee (5 \geq 4)) \wedge (\neg(14 + 23 = 38))$$

3 ist nicht kleiner 4, aber 5 ist größer 4, also ist die linke Doppelklammer wahr. Da auch 14+23 nicht 38 ist, wird mit dem NICHT die hintere Klammer wahr und die gesamte Aussage auch.

6. Aufgabe

(4 Punkte)

Hier eine act-Methode von Robita aus dem bekannten Roboterszenario:

```
public void act()
{
    if (!wandVorne())
    {
        bewegen();

        if (akkuAufFeld() && !wandVorne())
        {
            akkuAufnehmen();
        }
        else
        {
            System.out.println("Warum?!");
        }
    }
}
```

Beschreibe detailliert, welcher Vorgang abläuft, wenn du hier die play-Taste drückst:



Da vor Robita keine Wand ist, bewegt sie sich und nimmt den Akku auf. Danach geht sie wieder ein Feld weiter und nimmt den zweiten Akku auf. Danach geht sie wieder ein Feld weiter. Da hier kein Akku zu finden ist, sagt sie „Warum?!“, geht danach auf das nächste Feld, nimmt den Akku auf und bemerkt die Wand. Hier stoppt das Programm, denn es gibt keine ELSE-Bedingung.

7. Aufgabe

(8 Punkte)

Robita beginnt in einer Position, in der eine Mauer rechts von ihr ist. Robita soll entlang der Mauer gehen und in jeder Mauerlücke eine Schraube ablegen. Die Abfolge der Mauerlücken ist dabei unbekannt. Ein Beispiel:



Schreibe für Robita eine act-Methode, die bewirkt, dass sie alle zu Beginn in Bewegungsrichtung vorliegenden Mauerlücken mit einer Schraube „füllt“. Nutze Java- || Pseudo-Code und verwende

für Robitas Aktionen ausschließlich folgende im Roboterszenario mitgelieferten Methoden: **wandVorne()**, **bewegen()**, **dreheRechts()** und **schraubeAblegen()**.

Hier ist der Trick, mit dreheRechts zu beginnen. Dann prüft man mit wandVorne auf die Lücke. Ist da eine, wird der Akku abgelegt und es geht weiter. Ist da keine, wird mit 3dreheRechts und 1xbewegen das Programm beendet und von vorne begonnen.