**B02: Übungen zu Grundoperationen 'Binär':**

Sie lösen von jeder der folgenden Aufgaben 1.1 bis 1.6 mindesten 3 der jeweils 10 vorhandenen  
 Teilaufgaben und melden alle Ihre Probleme bzw. Unklarheiten spätestens bei der Besprechung!  
 Ihre Lösungen sollten dabei folgendes Beispiel '**1374 : 13 =** ' einer Musterlösung nicht unterbieten:

**1374 : 13 =** => 55E16 : D16 = => 0101010111102 : 11012 = 11010012

-1101

10000

-1101

001111

-1101

0010110

-1101

10012 Rest

**Aufgabe 2.1:** Addieren Sie binär die folgenden 3 Teilaufgaben schriftlich und detailliert!

**a)** 2 + 3 =

**b)** 9 + 19 =

**c)** 128 +127 =

**Aufgabe 2.2:** Subtrahieren Sie binär die folgenden 3 Teilaufgaben schriftlich und detailliert!

**a)** 26 - 11 =

**b)** 64 -32 =

**c)** 255 -118 =

**Aufgabe 2.3:** Multiplizieren Sie binär die folgenden 3 Teilaufgaben schriftlich und detailliert!

**a)** 4 · 4 =

**b)** 10 · 4 =

**c)** 15 · 14 =

**Aufgabe 2.4:** Dividieren Sie binär die folgenden 3 Teilaufgaben schriftlich und detailliert!

**a)** 30 : 6 =

**b)** 100 : 5 =

**c)** 100 : 7 =

**Aufgabe 2.5:** Verknüpfen Sie die beiden Binärzahlen: 010101012 und 111100002!

**a)** AND

**b)** OR

**c)** XOR

**Zusatzaufgabe 3:** Binär-Addierer in C#

Erstellen Sie mit C# einen Addierer, der Binärzahlen zusammenzählen kann.

Zu Beginn sollen erstmals 4-Bit Zahlen addiert werden.

Später kann das Programm sehr einfach auf grössere Zahlen umgebaut werden.

**Tipps:** - Lesen Sie die beiden Binärzahlen als String ein.

- Wandeln Sie den String in einen Array von char[] um.   
Nutzen Sie dazu die String-Methode ToCharArray().

- Um die Übersicht zu behalten, ist es hilfreich von Hand eine Tabelle zu erstellen, wo alle Möglichkeiten für die Summe und den Übertrag eingetragen sind. Es gibt 6 verschiedene Möglichkeiten.

Die Oberfläche könnte wie folgt aussehen

