



Aufgabenblatt 3

Ausgabe: 28.10., Abgabe: 04.11. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 3.1 (Punkte 20·1)

Darstellung negativer Zahlen: Geben Sie für jedes der folgenden 8-bit Bitmuster die fünf dezimalen Werte an, die sich bei deren Interpretation als

1. (positive) ganze Zahl im Dualsystem,
2. Betrag und Vorzeichen,
3. Exzess-127 Kodierung,
4. Einerkomplement,
5. Zweierkomplement

ergeben. Bei der Betrag-und-Vorzeichen-Kodierung sei das Vorzeichen (wie üblich) im MSB kodiert.

- (a) 0000 1101
- (b) 0110 0110
- (c) 1010 1010
- (d) 1110 0011

Aufgabe 3.2 (Punkte 5+5+15)

Jemand schlägt folgenden Algorithmus zur Bildung des Zweierkomplements einer Binärzahl A vor:

Gehe die Bits von A von rechts nach links durch.

- 1) Lasse alle Nullen stehen, bis man eine Eins findet.
- 2) Lasse auch diese Eins stehen.
- 3) Invertiere die restlichen Bits von A .

- (a) Bestimmen Sie nach dem Algorithmus aus der Vorlesung die Zweierkomplementdarstellung der Dezimalzahl -58 (8 Bit).
 (b) Liefert der oben vorgeschlagene Algorithmus das gleiche Ergebnis?
 (c) Begründen Sie, warum dieser Algorithmus in jedem Fall das richtige Ergebnis liefert.

Aufgabe 3.3 (Punkte 5+5+5+10)

Arithmetische Operationen im Dualsystem: Gegeben seien die beiden (positiven) Zahlen im Dualsystem $A = 101101011$ und $B = 1101$. Berechnen Sie Summe, Differenz, Produkt und Quotient von A und B im Dualsystem. Geben Sie dabei jeweils die einzelnen Rechenschritte an.

Aufgabe 3.4 (Punkte 5 + 5)

Arithmetische Operationen im Hexadezimalsystem: Gegeben seien die beiden Hexadezimalzahlen $M = B357$ und $N = 380C$. Berechnen Sie im Hexadezimalsystem $M + N$ und $M - N$. Geben Sie dabei jeweils die einzelnen Rechenschritte an.

Aufgabe 3.5 (Punkte 5+10+5)

Vorsicht mit unsigned Operationen: In der Standard-C Bibliothek liefert die Funktion `strlen()` die Anzahl der Zeichen in einer Zeichenkette `s`:

```
typedef unsigned int  size_t;

size_t strlen( char *s );
```

Das Resultat der Funktion ist also vom Datentyp `size_t` bzw. `unsigned int`. Den Datentyp `char *s`, also Pointer auf Zeichen, werden wir später in der Vorlesung noch gründlich kennenlernen. Die Details spielen an dieser Stelle noch keine Rolle.

Sie möchten eine Funktion schreiben, die die Länge von zwei Zeichenketten `s` und `t` vergleicht. Konkret soll die Funktion den Wert 1 zurückgeben, wenn Zeichenkette `s` länger ist als Zeichenkette `t`, und sonst den Wert 0. Wir nehmen an, dass der Vergleichsoperator `>` für zwei Operanden mit ihrem C Compiler entsprechend funktioniert. Ihr Freund schlägt folgende Implementierung vor:

```
#include <stdio.h>

int strlonger( char *s, char *t ) {
    return strlen(s) - strlen(t) > 0;
}
```

- (a) In welchen Fällen liefert diese Funktion ein falsches Ergebnis?
 (b) Erläutern Sie, warum es zu dem Fehler kommt.
 (c) Verbessern Sie den Code, so dass die Funktion korrekt funktioniert.