Hochschule Deggendorf Prof. Dr. Peter Jüttner	
Vorlesung: Grundlagen der Informatik	WS 2012
Übung 3	Termin 16.10.12

Zahlensysteme - Musterlösung

1. Umwandlung von Festkommazahlen in Binärdarstellung

Wandeln Sie die folgenden Festkommazahlen in die entsprechende Binärdarstellung um:

a.) 1.25

Umwandlung des Vorkomma und Nachkommaanteils getrennt:

Vorkommaanteil: $1_{10} = 1_2$

Nachkommaanteil wird so lange mit 2 multipliziert, bis der Nachkommateil zu Null wird. 0.25 * 2 = 0.5; 0.5 * 2 = 1 1 umgewandelt in Binärform ist wieder 1. Diese muss noch um 2 Stellen nach rechts als Nachkommateil geschoben werden (da mit 2*2 multipliziert wurde)

D.h. $0.25_{10} = 0.01_2$

Vor- und Nachkommateil zusammengesetzt ergibt 1.25₁₀ = 1.01₂

Alternative Betrachtung: $0.25 = \frac{1}{4} = 1^{2^{-2}}$

b.) 1.625

Umwandlung des Vorkomma und Nachkommaanteils getrennt:

Vorkommaanteil: $1_{10} = 1_2$

Nachkommaanteil wird so lange mit 2 multipliziert, bis der Nachkommateil zu Null wird. 0.625 * 2 = 1.25; 1.25 * 2 = 2.5; 2.5 * 2 = 5; 5 umgewandelt in Binärform ist wieder 101. Diese muss noch um 3 Stellen nach rechts als Nachkommateil geschoben werden (da mit 2*2*2 multipliziert wurde) D.h. $0.625_{10} = 0.101_2$

Vor- und Nachkommateil zusammengesetzt ergibt 1.625₁₀ = 1.101₂

Alternative Betrachtung: $0.625 = 5.8 = 5^{2-3}$

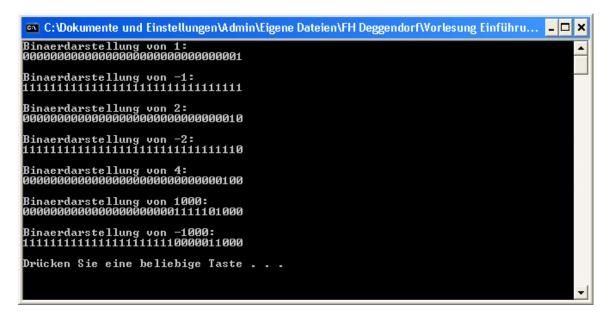
2. Binärdarstellung von ganzen Zahlen

Überlegen Sie, wie die unten angegebenen Zahlen binär dargestellt werden (die negativen Zahlen im 2-Komplement)

Implementieren Sie ein C-Programm, dass mit Hilfe der ganz unten angegebenen C-Funktion Print32BitLine diese Zahlen bitweise ausgibt. (die Funktion finden Sie auch auf dem Laufwerk)

Vergleichen Sie Ihre Ergebnisse mit der vom Programm erzeugten Ausgabe.

- a.) 1
- b.) -1
- c.) 2
- d.) -2
- e.) 4
- f.) 1000
- g.) -1000



3. Binärdarstellung von Gleitkommazahlen

Nutzen Sie die oben angegebene C-Funktion zur Ausgabe der folgenden Gleitkommazahlen

- a.) 1.0
- b.) 2.0
- c.) -1.0
- d.) -2.0
- e.) 1e10
- f.) -1e10

4. Mittels der Shift-Operatoren (<<, >>) lassen sich in C Bitfolgen um eine bestimmte Anzahl von Stellen nach links oder rechts schieben, z.B.

```
    int i = 5; i = i<<2; /* Die Variable i wird um zwei Stellen nach links geschoben */</li>
    int j = 100; j = j>>6; /* Die Variable j wird um 6 Stellen nach rechts geschoben */
```

Überlegen Sie, welches Ergebnis die folgenden Shift-Operationen haben. Wandeln Sie dazu die Dezimalwerte in Binärzahlen um. Probieren Sie sie in einem C-Programm aus und Vergleichen Sie Ihre Ergebnisse mit Ihren Überlegungen.

Hinweis: Rechtsshifts von vorzeichenbehafteten Zahlen ziehen das Vorzeichen nach (sonst 0)

```
a.) int i = 10; i = i << 4;</li>
b.) int i = 12; i = i >> 3;
c.) int i = -5; i = i << 4;</li>
d.) int i = -5; i = i >> 4;
e.) int i = -5; i = (unsigned int) i <<4;</li>
f.) int i = -5; i = (unsigned int) i >>4;
```

```
void Print32BitLine(unsigned long ui)
{    int i;
    for(i=0; i<32; i++)
        {        if(ui & (0x80000000>>i))
             printf("1");
        else
            printf("0");
        }
        printf("\n");
}
```