Übungsaufgaben A08 – A10
Programmierung in C, Bitlevel-Operationen

A08 Erzeugen von Bitfolgen auf Datenworten

Erzeugen Sie in einem C-Programm auf Variablen die unten angegebenen Bitfolgen! Geben Sie die danach Variablenwerte als Bitfolge und als Hexadezimaldarstellung auf der Konsole aus. Zur Ausgabe der Bitfolgen können Sie eine der Funktionen bitwise_display_uchar(), bitwise_display_ushort(), bitwise_display_uint() aus dem Beispielprogramm bitlevel2_basidatentypen.c benutzen, bzw. die Funktionen bitwise_display_unit8(), bitwise_display_uint16(), bitwise_display_uint32() aus dem Programm bitlevel2_stdint.c. Für die Hexadezimaldarstellung gibt es die printf()-Umwandlung %x.

```
0000 0000
0001 1100
1100 1111
1010 1111 1111 1010
0000 0000 1000 1110 0111 0001 0000 0000
```

Hier sind die Bitfolgen zeilenweise angegeben.
Die Darstellung mit 4-stelliger Gruppierung der Bits dient hier der besseren Lesbarkeit.

A09 Rechnen mittels Bitoperationen

Benutzen Sie in den Teilaufgaben a) und b) Bitlevel-Operationen für die Berechnungen. Nutzen Sie Variablen des Typs unsigned int (32 Bit), bzw. uint32_t aus den Typen, die durch stdint.h bereitgestellt werden.

Teilaufgaben:

a) Belegen Sie eine Variable wert mit einem Startwert, beispielsweise mit 15! Berechnen Sie alle Werte, die durch schrittweise Verdoppelung des Werts entstehen, solange die höchstwertige Bitstelle 0 ist! Nutzen Sie Bitlevel-Operationen und geben Sie nach jeder Veränderung den Wert im Dezimalsystem und als Bitfolge aus!

b) Belegen Sie eine Variable wert mit einem großen Startwert, beispielsweise mit 478761! Berechnen Sie schrittweise die Werte, die durch aufeinanderfolgendes Dividieren durch 2 entstehen, solange Null noch nicht erreicht wird! Geben Sie nach jeder Veränderung den Wert als Dezimalzahl und als Bitfolge aus!

A10 Auslesen, Setzen, Rücksetzen, Invertieren einzelner Bits

Schreiben Sie jeweils eine Funktion, die auf einem 32-Bit-Bereich (eine Variable des Typs unsigend int, bzw. uint32_t aus stdint.h)

- prüft, ob Bit i (i zwischen 0 und 31 von rechts beginnend aufwärts) gesetzt ist.
- das Bit i setzt.
- das Bit i zurücksetzt.
- das Bit i invertiert (aus 1 soll 0 werden und umgekehrt).

Testen Sie Ihre Funktionen durch ein Hauptprogramm in geeigneter Weise!