

Musterlösung der 7. Übung

1. Vollständiger Quelltext:

```
int minimum(int *feld, unsigned int anzahl)
{
    int i;
    int minimum = feld[0];
    for(i=1; i<anzahl; i++)
        if(feld[i] < minimum) minimum = feld[i];
    return minimum;
}
```

2. Vollständiger Quelltext:

```
#include <memory.h>

struct Buch
{
    char *titel;
    char *autor;
    char *kurzinfo;
    int nummer;
    struct Buch *next;
};

int main()
{
    int i;
    struct Buch *erstes = (struct Buch *) malloc(sizeof(struct Buch));
    struct Buch *aktuell = erstes;

    aktuell->nummer = 1;

    for(i=1; i<1000; i++)
    {
        aktuell->next = (struct Buch *) malloc(sizeof(struct Buch));
        aktuell->next->nummer = i + 1;
        aktuell = aktuell->next;
    }

    aktuell->next = NULL; /* Liste abschließen */

    return 0;
}
```

3. Vollständiger Quelltext:

```
void setze_bit1(int *zahl)
{
    *zahl = *zahl | 2;
    // Alternativ: *zahl |= 2;
}
```

4. Vollständiger Quelltext:

```
int ist_durch_4_teilbar(int zahl)
{
    // Wenn Bit 0 oder 1 gesetzt ist,
    // ist die Zahl NICHT durch 4
    // teilbar.
    if(zahl & 3) return 0;
    else return 1;

    // Kürzere Alternative:
    // return (zahl & 3) ? 0 : 1;

    // Noch kürzer:
    // return !(zahl & 3);
}
```

5. Vollständiger Quelltext:

```
void teile_durch_2(int *zahl)
{
    *zahl = *zahl >> 1;

    // Kürzer:
    // *zahl >>= 1;
}
```

6. Vollständiger Quelltext, inkl. main() zum Testen:

```
#include <stdio.h>

void rueckwaerts_ausgeben(char *text)
{
    if(*text) // Wenn Ende ('\000') noch nicht erreicht
    {
        // Erst das nächste Zeichen auswerten!
        rueckwaerts_ausgeben(text + 1);
        // DANN das aktuelle ausgeben
        printf("%c", *text);
    }
}
```

```
int main() // Test, Test...
{
    char *a = "Hallo, Welt!";

    rueckwaerts_ausgeben(a);
    printf("\n"); // Zeilenende-Zeichen

    return 0;
}
```