Hochschule Deggendorf Dr. Peter Jüttner	
Vorlesung: Einführung in die Programmierung	WS 2012
Übung 4	Termin 23.10.12

Ein-/Ausgabe und Operatoren - Musterlösung

1. **Ein-/Ausgabe von Zeichenketten.**

   Schreiben Sie ein Programm, das nacheinander einen Vornamen, Nachnamen, eine Adresse und einen Wohnort einliest und anschließend in Form einer Adresse auf einem Briefkopf wie unten angegeben auf dem Bildschirm wieder ausgibt.

   Hans Mustermann
   Zufallsstrasse 17
   12345 Beispielort

   Hinweise:

   a. Zum Einlesen einer Zeichenkette benötigen Sie eine Variable vom Typ einer Zeichenkette. In C gibt es diesen Typ explizit nicht. Er wird implizit durch ein Feld von Buchstaben (char) mit einer bestimmten Länge ersetzt.

   b. Ein Feld vom Typ char, das z.B. 20 Buchstaben aufnehmen kann, legen Sie mit folgender Variablendefinition an:

      char bf[20] **; /*** Feld von 20 Buchstaben */

      Das Einlesen von Daten auf diese Variable namens bf geschieht durch folgenden Aufruf der scanf-Funktion:

      scanf(“%s“,bf);

      Achtung, die Verwendung des Adressoperators (&) entfällt bei Zeichenketten!

      Achtung: scanf liest Zeichenketten nur bis zu einem Leerzeichen!

Achtung: Als letztes Zeichen wird beim Einlesen automatisch ein Endezeichen ’\0’ eingefügt.

c. Achten Sie darauf, Ihre Variable für die einzulesenden Daten mit einer ausreichenden Länge zu definieren!

d. Alternativ können Sie für Zeichenketten auch die Funktion gets() verwenden mit folgenden Aufruf:

   gets(bf);

   gets(bf) liest bis zum Zeilenende.

   Dabei ist vor dem ersten Einlesen mit gets der Eingabepuffer mittels fflush(stdin); zu leeren.

Die folgende Lösung arbeitet mit der Einlesefunktion gets(), die eine Zeichenfolge bis zum Zeilenende (Return-Taste) einliest. Dabei werden, im Gegensatz zu scanf(...) Leerzeichen mit eingelesen. gets() benötigt keine Formatangabe, da nur Zeichenketten eingelesen werden.
Bei einer Lösung mittels scanf(...) muss Vor- und Nachname in getrennten Variablen abgespeichert werden, ebenso Strasse und Hausnummer, wie auch Postleitzahl und Ort, da bei der Eingabe bis zu einem Leerzeichen gelesen wird und nicht bis zum Zeilenende.

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
  char name[40];
  char strasse[40];
  char ort[40];

  fflush(stdin); /* Löschen Eingabepuffer */
  printf ("Bitte Vor- und Nachname eingeben (max. 39 Zeichen)\n");
  gets(name);
  printf ("Bitte Strasse und Hausnummer eingeben (max. 39 Zeichen)\n");
  gets(strasse);
  printf ("Bitte Plz. und Ort eingeben (max. 39 Zeichen)\n");
  gets(ort);

  printf("\n%s\n", name);
  printf("%s\n", strasse);
  printf("%s\n", ort);
```

```
  system("PAUSE"); /* nicht für Visual Studio notwendig */
}
```

2. **Shift-Operatoren**
   Schreiben Sie ein Programm, das zwei Zahlen (a ganze Zahl und n natürliche Zahl) einliest und mittels Shift-Operationen $a*2^n$ und $a*2^{-n}$ berechnet.
   Probieren Sie Ihr Programm für verschiedene Zahlen aus
   Stellen Sie fest, wo die „Grenzen“ liegen, d.h. wann die Ergebnisse nicht mehr korrekt sind.

```
#include <stdio.h>
#include <stdlib.h>

int main (void)
{
  int a=0;
  unsigned int n = 1;
  while (n != 0)
  { printf("Bitte Zahl a eingeben:\n");
    scanf("%d",&a);
    printf("Bitte Zahl n eingeben:\n");
    scanf("%d",&n);

    printf("%d um %d Stellen nach links geshiftet (entspricht a * (2 hoch
n)  : %d\n", a, n, a<<n);
    printf("%d um %d Stellen nach rechts geshiftet (entspricht a * (2 hoch -
n): %d\n", a, n, a>>n);
  };
  system("PAUSE");
}
```

3. **Bitweise Operatoren**
   Schreiben Sie ein Programm, das eine Zahl einliest, alle Bits der Zahl invertiert und die Zahl anschließend wieder ausgibt.
   Was stellen Sie fest? Begründen Sie Ihre Feststellung

```
#include <stdio.h>
#include <stdlib.h>

int main (void)
{
  int a=1;
  while (a != 0)
  { printf("Bitte Zahl a eingeben:\n");
```

```
    scanf("%d",&a);

    printf("%d alle Stellen invertiert ergibt %d \n", a, ~a);

  };
  system("PAUSE");
}
```

Das Invertieren der Bits einer ganzen Zahl ergibt das Negative der Zahl -1. Grund ist die Tatsache, dass intern das Negative einer Zahl durch 2-er-Komplementbildung dargestellt wird. 2-er-Komplementbildung heißt Invertieren aller Bits und 1 draufaddieren.