



Betreutes Programmieren 2

Aufgabe 2 ** (*Operatoren, Eingabe und Ausgabe*)

Heute werden Sie sich ein wenig mit den Operatoren und den Möglichkeiten der formatierten Ausgabe in C beschäftigen. Als erste Anlaufstelle für Fragen sollten Ihnen die folgenden zwei Internetseiten (bei Letzterer nur 2.1 und 2.2 für die hier gestellten Aufgaben relevant) dienen:

- http://de.wikibooks.org/wiki/C-Programmierung:_Einfache_Ein-_und_Ausgabe
- http://de.wikibooks.org/wiki/C-Programmierung:_Ausdr%C3%BCcke_und_Operatoren

Die Aufgabe besteht darin, ein C-Programm zu erstellen und die formulierten Anweisungen entsprechend in C umzuformulieren. Das Programm wird dabei in Worten beschrieben, wobei jeder Punkt in genau eine Zeile zu überführen ist.

Beispielsweise soll

- Deklarieren Sie sich drei Variablen `laenge`, `breite` und `hoehe` vom Typ `int`.

in den Code

```
int laenge, breite, hoehe;  
überführt werden.
```

Es sollen nach jeder Anweisung alle aktuellen Variablenwerte in der Form `Variablenname=Wert` ausgegeben werden.

Erstellen Sie für jede Teilaufgabe jeweils eine C-Datei mit einer eigenen `main()`-Funktion. Kompilieren Sie Ihre Programme mit den Compilerschaltern `-ansi -pedantic -Wall -Wextra`.

a) Datentyp `int`

Erstellen Sie eine Datei `bp02a.c`. Bei der Ausgabe sollen die Werte von `i` und `j` immer genau vier Zeichen belegen, wobei bei `i` mit Leerzeichen und `j` mit Nullen aufgefüllt werden soll.

- Deklarieren Sie drei Variablen `i`, `j` und `k` vom Typ `int`, wobei Sie `i` mit dem Wert 42 und `j` mit dem Wert 9 initialisieren. Erklären Sie den Wert von `k`!
- `k` soll nun der Wert von `i` subtrahiert um den Wert 21 zugewiesen werden.
- `k` soll nun der Rest der Division des Werts von `i` mit dem Wert von `j` zugewiesen werden.

- Multiplizieren Sie `k` mit `j` und weisen Sie das Ergebnis `i` zu.
- Der Benutzer soll unter Verwendung einer einzigen `scanf`-Anweisung zur Eingabe von zwei ganzen Zahlen aufgefordert werden, die ohne Leerzeichen nur durch einen Doppelpunkt getrennt in einer Zeile eingegeben werden sollen. Dabei soll die erste Zahl in `i`, die zweite Zahl in `j` und der Rückgabewert dieser `scanf`-Anweisung in `k` gespeichert werden. Erklären Sie, was es mit den gespeicherten Werten bei unterschiedlichen Falscheingaben auf sich hat.

b) Datentyp `int`

Erstellen Sie eine Datei `bp02b.c`. Bei der Ausgabe soll der Wert von `r` mit der vordefinierten Anzahl an Nachkommastellen, der Wert von `s` immer mit genau zwei Nachkommastellen und der Wert von `t` immer mit Fließkommadarstellung erscheinen.

- Deklarieren Sie drei Variablen `r`, `s` und `t` vom Typ `double`, wobei Sie `r` mit dem Wert 9.61, `s` mit dem Wert 15.0 und `t` mit dem Wert 0.1 initialisieren.
- `s` soll nun das Zwanzigfache von `t` zugewiesen werden.
- Addieren Sie `r` und `t`. Speichern Sie dann das Ergebnis in `t`.
- Speichern Sie das Hundertfache des negativen Werts von `r` in `t` ab.
- In `s` soll nun die Wurzel aus `r` gespeichert werden. Dafür wird in der Standard-Bibliothek `math.h` eine passende Funktion bereitgestellt, die Sie selbst recherchieren sollen.

c) Datentyp `char`

Erstellen Sie eine Datei `bp02c.c`. Geben Sie jeweils `c`, `d` und `e` als `int`- und als `char`-Wert aus.

- Deklarieren Sie drei Variablen `c`, `d` und `e` vom Typ `char`, wobei Sie `c` mit dem Wert `'b'`, `d` mit dem Wert `'1'` und `e` mit dem Wert 69 initialisieren.
- `e` soll durch den Wert 3 dividiert werden. Wie erklären Sie die Ausgabe von `e`?
- In `d` soll das Ergebnis der Addition von `c` und `e` gespeichert werden.
- Erhöhen Sie den Wert von `e` um das Zeichen `'N'`.
- Wandeln Sie das in `c` gespeicherte Zeichen in einen Großbuchstaben um. Dazu gibt es eine geeignete Funktion in der Standard-Bibliothek `ctype.h`, deren Verwendung Sie selbst nachschlagen sollen.
- Geben Sie nun zum Abschluss der letzten Teilaufgabe die Variablen `c`, `d` und `e` direkt hintereinander nur als Zeichen (ohne Leerzeichen dazwischen) gefolgt von drei Ausrufezeichen aus.