Übungsblatt 2 Formatierte Ausgabe

Lernziele:

- Ausgabe via printf-Funktion vertiefen
- das Formatieren von Text erlernen
- den Umgang mit Formatbuchstaben üben

Legende:

Lektüre, die vor der Übung gelesen werden muss

Fragen/Aufgaben, die vor der Übung zu bearbeiten sind

❖ Aufgaben, die in der Übung zu bearbeiten sind

In der vergangenen Übung wurde bereits die Funktion printf vorgestellt, die eine formatierte Ausgabe von Text aus einem C-Programm ermöglicht. Die Möglichkeiten dieser Funktion wollen wir uns in dieser Übung näher anschauen. Die Funktion ist wie folgt deklariert:

```
printf(const char * format,...);
```

Diese **Deklaration** ist bereits in der Datei stdio.h hinterlegt und wird durch **#include** <stdio.h> in jedem C-Programm, das die Funktion nutzen will, eingebaut. D.h. die **Deklaration darf nicht** noch einmal in Ihrem C-Programm hingeschrieben werden.

Ein passender Aufruf der Funktion printf sieht z.B. so aus:

```
printf("Hallo Welt!\n");
```

Dabei wird die auszugebende Zeichenkette, also Hallo Welt, im Datentyp (dieser Begriff wird später vertieft) passend für format übergeben. Die Zeichenkette, auch als auszugebender Text bezeichnet, muss dabei immer mit "beginnen und mit "enden.

Die **Tabelle 1** im Anhang auf S. 7 zeigt eine Liste der Formatbuchstaben, mit der in der **printf**-Funktion verschiedene Zahlenformate ausgeben werden können. In **Tabelle 2** auf S. 8 sind die Steuerzeichen dargestellt. Mit Hilfe dieser Steuerzeichen lassen sich Texte schnell und einfach formatieren, wie das mittels \n für eine neue Zeile im obigen Hallo-Welt-Beispiel bereits gezeigt wurde.

Prof. Dr.-Ing. J. Abke Simón Rodriguez Perez Patricia Stegmann

TH Aschaffenburg Informatik I

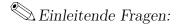
2.1 Vorbereitung zur Übung

Die Vorbereitungen und Fragen sind vor dem Übungstermin zu bearbeiten. Sie sind als Hilfen gedacht, um Ihnen die Aufgaben, die an den Übungsterminen selbst zu bearbeiten sind, zu erleichtern.

 \mathbb{L} Lesen Sie die Kapitel 4.2 und 5.20 im Buch "C von A bis Z" von Jürgen Wolf!

Prof. Dr.-Ing. J. Abke Simón Rodriguez Perez Patricia Stegmann

TH Aschaffenburg Informatik I



Welche Datei müssen Sie zuerst mit dem #include-Befehl in Ihr Projekt einbinden, um die printf-Funktion in Ihrem Programm benutzen zu können?

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

🛇 Eine kleine Übung zum Aufwärmen

Geben Sie die Zahl 10 jeweils als Dezimal-, Oktal- und Hexadezimalzahl aus.

Wählen Sie hierzu zuerst aus der Tabelle 1 (siehe S. 7) die für die Aufgabenstellung passenden Formatbuchstaben aus und schreiben Sie diese für die drei Zahlentypen in den folgenden Kasten.

```
%d oder %i für die Dezimalzahlen
%o für Oktalzahlen
%x oder %X für kleine/große Buchstaben bei der Ausgabe der Hexadezimalzahl
```

Die Programmidee sollten Sie danach hier eintragen.

```
#include <stdio.h>
int main(void)
{
    int zahl = 10;
    printf("Dez: %d \n", zahl);
    printf("Okt: %o \n", zahl);
    printf("hex: %x \n", zahl);
    printf("HEX: %X \n", zahl);
```

```
return 0;
```

2.2 Ausgabe der Zahlen 8 bis 16 in verschiedenen Formaten

Aufgabe:

Lassen Sie sich die Zahlen 8 bis 16 auf der Konsole ausgeben. Die Zahlen sollen dabei als Dezimal-, Oktal- und Hexadezimalzahlen ausgegeben werden. Nutzen Sie hierzu die entsprechenden Formatbuchstaben, sie Sie in der vorherigen Übung ermittelt haben.

Für die einzelnen Zahlenformate soll dabei eine neue Zeile begonnen werden.

HIER BENÖTIGTE FUNKTIONEN:

printf("Textbeispiel");

2.3 Formatierte Textausgabe

Aufgabe:

Geben Sie nachfolgenden Text auf der Konsole aus.

Hochschule
Aschaffenburg
Wuerzburger
Strasse 45
63743 Aschaffenburg

Gebaeude: 40 Raum: 130

Wählen Sie aus der Tabelle 2 (siehe Anhang) die für die Aufgabenstellung passenden Steuerzeichen aus! Begründen Sie Ihre Auswahl!

\n -> Zeilenumbruch \t -> Tabulator

 \clubsuit Schreiben Sie nun das Programm, in dem Sie die gewonnenen Erkenntnisse umsetzen!

HIER BENÖTIGTE FUNKTION:

• printf("Textbeispiel");

2.4 Formatierung der Genauigkeit und Stellenangaben

Für die Ausgabe von Zahlen existieren eine Reihe von Zusatzoptionen, die zwischen % und Formatierungsbuchstaben geschrieben werden. Diese Zusatzoptionen wirken sich u.a. auf rechts-, linksbündige Ausgabe mit/ohne führenden Nullen, die Anzahl der Nachkommastellen (bei Fließkommazahlen) aus.

Recherchieren Sie zunächst unter

http://www.tutorials.at/c/03-dateneingabe-ausgabe.html, wie welche Zusatzoption in C realisiert wird.



Überlegen Sie, welche Formatbuchstaben Sie für die Zahlen 4711 und 0.24124benötigen.

Zahl	Formatbuchstabe
4711	%d
0.24124	%f

Vervollständigen Sie nun folgende Tabelle:

Zusatzoption	4711	0.24124
rechtsbündig	% 6d	% 10f
linksbündig	% -6d	%-10f
mit Vorzeichen	% +6 d	% +10 f
mit Vorzeichen links- bündig	%-+6d	% -+10f
mit führender Null	% 06d	% 010f
mehrere Stellen einer Zahl (zB. 8) angeben	% 8d	% 8f
mit Vorzeichen, mehrere Stellen und führender Null	% +08d	% +010f
mehrere Nachkommastellen angeben (zB. 8)	-	% .8f

 $[\]mbox{\Large $\rlap \ $\rlap $\ $\rlap \ $ $ $}$ Lassen Sie sich nun die beiden Zahlen nache
inander mit den verschiedenen Zusatzoptionen auf der Konsole ausgeben.

Anhang

Tabelle 1: Liste der Formatbuchstaben

Format	Argumenttyp	Erläuterung	Beispiel
С	char oder int	einzelnes Zeichen	printf("%c", 'A'); druckt A
d	int	dezimal	printf("%d", -45); druckt -45
i	int	dezimal	printf("%i", -45); druckt -45
u	unsigned int	dezimal ohne Vorzeichen	printf("%u", 3); druckt 3
О	unsigned int	oktal ohne Vorzeichen	printf("%o", 8); druckt 10
X	unsigned int	hexadezimal 0-9abcdef	printf("%x", 11); druckt b
X	unsigned int	hexadezimal 0-9ABCDEF	printf("%X", 11); druckt B
f	float	Gleitpunktzahl	printf("%f", 5.234);
			druckt 5.234000
e,E	float	Exponentenschreibweise	printf("%e", 0.000005);
			druckt 5.000000e-6
g,G	float	das kürzere von f und e	
S	string	Ausgabe bis '\0'	<pre>printf("%s", "string");</pre>
			druckt string
ld	long int	dezimal	printf("%ld", -45456256432);
			druckt -45456256432
lf	double	Gleitpunktzahl	
		doppelt genau	
le, lE	double	Exponentenschreibweise	
		doppelt genau	
hd	short int	dezimal, Short-Int	
hx	short int	hexadezimal, Short-Int	

Tabelle 2: Liste der Steuerzeichen

\b	Backspace (ein Zeichen zurück)
\t	Tabulator (Horizontaler Zeilenvorschub, meist um 8 Zeichen)
\f	Seitenvorschub
\n	Neue Zeile
\r	Wagenrücklauf
	(an den Anfang d. akt. Zeile)
\v	Vertikales Tabulatorzeichen
\'	Hochkomma
\"	Anführungsstriche
//	Backslash