2. Übungsblatt: Programmierpraktikum II (SS 2003)

Abgabe: 19. Mai 2003

Bemerkungen:

- Es sind alle Abgabehinweise auf der Übungswebseite zu beachten!
- Als Programmbibliothek ist nur `stdio.h` zugelassen.
- Hinweise zum Programmieren von Ein- und Ausgabe finden sich im Anhang dieses Übungsblattes.

Aufgabe 1: Operatoren in C

(6 Punkte, Abgabe nur schriftlich)

Schreibe für jede Teilaufgabe a) bis c) ein kleines C-Programm, das den Operator

a) `a++`

b) `|`

c) `=`

verwendet. Das C-Programm soll dabei so gestaltet sein, dass sich sein Output ändert, wenn man den oben genannten Operator ersetzt durch

a) `++a`

b) `||`

c) `==`

Das Programm soll noch immer ohne Fehlermeldung compilieren (Warnungen sind aber diesmal ausdrücklich erlaubt). Erläutere kurz, wie sich die Programme verhalten und warum es zu unterschiedlichem Output kommt!

Aufgabe 2: Euro-Umrechner

(6 Punkte, Abgabedatei: euro.c)

Schreibe ein C-Programm, das DM in Euro umrechnet oder umgekehrt. Dazu liest es zunächst den Betrag und dann die Währung (0 für DM oder 1 für Euro) von der Standardeingabe. Dann rechnet es den korrespondierenden Wert (entweder Euro oder DM) aus und gibt ihn an die Standardausgabe zurück. Falls als Währung weder 0 noch 1 angegeben wurden, soll das Programm einen Fehler melden und sich beenden.

Tipp: Das Programm benötigt eine `if`-Anweisung, die genauso funktioniert wie bei Java.

Aufgabe 3: Zinseszinsen

(8 Punkte, Abgabedatei: zins.c)

Schreibe ein C-Programm, das eine Zinseszinstabelle erstellt. Wir betrachten dabei einen Anleger, der 1000 Euro Startkapital besitzt. Die Tabelle soll in Zeile i und Spalte j angeben ($i, j = 1, \ldots, 10$), welchen Betrag der Sparer bei i Prozent Zinsen nach j Jahren ausgezahlt bekommt.

Tipp 1: Du kannst die aus Java bekannte `for`-Anweisung benutzen. Beachte aber, dass in C alle verwendeten Variablen zu Beginn der Funktion (hier z.B. `main()`) deklariert werden müssen!

Tipp 2: Benutze als Ausgabeanweisung den Befehl

```
printf("%8.2f",s);
```

auf diese Weise erhältst du genau zwei Hinterkommastellen und eine ordentliche Formatierung der Ausgabe.

Ein-/Ausgabe in C

Die Befehle, die in C die Ein-/Ausgabe steuern, sind in der Headerdatei `stdio.h` deklariert. Diese muss (so wie im Beispielprogramm "Hello World" gezeigt) zu Beginn des Quellcodes mit einer `#include`-Anweisung eingebunden werden.

Ausgabe: Wir benutzen im Rahmen dieses Kurses zunächst nur die Ausgabefunktion `printf`. Eine besonders einfache Anwendung dieser Funktion ist ebenfalls im "Hello World"-Programm zu sehen: Man übergibt der Funktion einfach den String, den man ausgeben möchte, also z.B.

```
printf("Dies ist ein Output-String.\n");
```

Die einzige Besonderheit hier ist die Zeichenfolge `\n` - sie steht lediglich für einen Zeilenumbruch.

Schwieriger wird das Ganze, wenn man den Wert von Variablen ausgeben will. Dazu muss man zunächst in den String einen Platzhalter für die auszugebende Variable einfügen und dann hinter dem String die Variable selbst. Ein Beispiel:

```
int zahl = 12;
printf("Es gibt %i Monate.\n", zahl);
```

Hier ist also `%i` der Platzhalter für einen Integer-Wert. Analog ist z.B. `%f` der Platzhalter für einen Float-Wert. Sollen mehrere Variablen ausgegeben werden, so geht man genauso vor:

```
int zahl, quadrat;
zahl = 5;
quadrat = zahl * zahl;
printf("Das Quadrat von %i ist %i.\n", zahl, quadrat);
```

Hier wird also offensichtlich der erste Platzhalter durch die erste Variable und der zweite Platzhalter durch die zweite Variable ersetzt.

Eingabe: Die Eingabe funktioniert ähnlich wie die Ausgabe. Wir beschränken uns hier zunächst auf den Befehl `scanf`. Auch hier wird wieder ein String verwendet, der sogenannte *Formatkontrollstring*. Dieser enthält ähnliche Platzhalter für die Eingabe wie beim `printf`-Befehl, also z.B. `%i` für Integer- oder `%f` für Float-Werte. Bsp.:

```
int zahl;
printf("Gib eine ganze Zahl ein!\n");
scanf("%i", &zahl);
```

Beachte hier vor allem, dass als Argument nicht `zahl`, sondern `&zahl` verwendet wird. Für eine Erklärung dieses `&`-Zeichens müssen wir uns allerdings noch ein paar Wochen gedulden - im Augenblick genügt es zu wissen, dass es da hingehört.

Weiterführendes: Wer sich mit solch spärlichen Erklärungen nicht abspeisen lassen will, der kann natürlich gerne ein Buch zur Hand nehmen - die Befehle `printf` und `scanf` sind in beiden Büchern zur Vorlesung ausführlich erklärt. Andernfalls müsst ihr noch ein paar Wochen warten, bis in der Vorlesung das Kapitel "Ein- und Ausgabe" behandelt wird.