Prof. Dr. Wolfgang Effelsberg
Dirk Stegemann Mannheim, den 02.05.2005

Vorlesung **Programmierkurs 2**, Sommersemester 2005

Übungsblatt 3

Abgabe spätestens am Montag, 09. Mai 2005, um 13.45 Uhr.

Aufgabe 1: Präzedenzregeln

(6 Punkte, Abgabe nur schriftlich)

Welche Werte haben die Variablen `a`, `b` und `c` nach Ausführung des folgenden C-Codes? Begründe deine Lösung, indem du für jeden Ausdruck die Auswertungsreihenfolge der Operationen durch entsprechende Klammerung deutlich machst.

```
int a=0;
int b=0;
int c=0;

a = b | 1 * 7 >> 4 % 3;
b = a == c ^ 0xff + 7 / 5;
c = 4 > 012 - 8 < 15 || a;
```

Aufgabe 2: Zinseszinsen

(7 Punkte, Abgabedatei `zinseszinsen.c`)

Schreibe ein C-Programm, das eine Zinseszinstabelle erstellt. Das Programm soll den Benutzer nach dem gewünschten Startkapital fragen und daraufhin eine Tabelle ausgeben, die in Zeile i und Spalte j ($i, j \in \{1, \ldots, 10\}$) angibt, welcher Betrag dem Sparer bei i Prozent Zinsen nach j Jahren ausgezahlt würde.

Hinweis: Zur Formatierung der Ausgabe kannst du beim Aufruf der `printf`-Funktion Platzhalter der Form `%`n`.`m`f` verwenden, die die entsprechende Zahl rechtsbündig mit m Nachkommastellen in einem insgesamt n Zeichen breiten Feld ausgeben. Beispielsweise erzeugt die Anweisung `printf("12345678\n%8.2f\n", 37.25)` die Ausgabe

```
12345678
   37.25
```

Aufgabe 3: Mehr-oder-Weniger

(7 Punkte, Abgabedatei `mehrwenig.c`)

(i) (6 Punkte) Schreibe eine C-Implementation für das Mehr-oder-Weniger Spiel. Dabei zieht das Programm zufällig eine Zahl $i \in \{1, \ldots, 100\}$, und der Benutzer darf so lange Tipps abgeben, bis er die Zahl erraten hat. Nach jedem falschen Tipp gibt das Progamm aus, ob die gesuchte Zahl größer oder kleiner ist als die getippte Zahl. Am Ende soll das Programm ausgeben, wie viele Tipps der Benutzer benötigt hat, um die Zahl zu erraten.

Die Ausgabe des Programms sollte in etwa wie auf der nächsten Seite dargestellt aussehen.

```
Dein Tipp? 42
Meine Zahl ist groesser als 42.
Dein Tipp? 56
Meine Zahl ist kleiner als 56.
Dein Tipp? 49
Richtig!
Du hast 3 Versuch(e) gebraucht, um die Zahl zu erraten.
```

(ii) (1 Punkt) Wie kann der Benutzer die gesuchte Zahl mit möglichst wenigen Versuchen erraten?

Hinweis: Zur Erzeugung von Zufallszahlen kannst du den Standard-Zufallszahlengenerator von C verwenden, der in der Datei `stdlib.h` beschrieben ist. Damit er nicht bei jedem Programmaufruf dieselben Zufallszahlen liefert, muss der Zufallszahlengenerator initialisiert werden. Diese Initialisierung wird oft mit Hilfe der aktuellen Uhrzeit vorgenommen. Dazu muss man noch die Datei `time.h` mit einbinden und die Anweisung `srand(time(0))` zu Beginn des Programms ausführen.

Die Funktion `rand()` liefert eine Pseudo-Zufallszahl im Bereich zwischen 0 und einem Wert `RAND_MAX`, der ebenfalls in `stdlib.h` definiert ist. Mittels `rand()% n` kann man Zahlen zwischen 0 und $n - 1$ erzeugen.