



## 2. Tutoriumsblatt zur „Algorithmischen Diskreten Mathematik“

### Übung am Computer

Heute wollen wir uns vor allem mit Kontrollstrukturen, wie Schleifen und Alternativen, beschäftigen. Außerdem lernen Sie neues über das Einlesen von der Tastatur und die Ausgabe am Bildschirm. Falls Sie noch nicht alle Aufgaben des letzten Blattes bearbeitet haben, schauen Sie sich zunächst diese nochmal an.

#### Aufgabe C11 (Warm-up: Schleifen)

Grundsätzlich kann man jede `for`-Schleife auch als `while`-Schleife schreiben und umgekehrt. Schreiben Sie die Schleife

```
for (i = 0; i < 10; i++) printf ("%d\t",i);
```

als `while`-Schleife.

#### Aufgabe C12 (Binomialkoeffizient)

Mit Hilfe der Funktion `scanf()` (siehe auch `man scanf`) kann wie folgt eine Zahl von der Tastatur eingelesen werden:

```
#include <stdio.h>
...
int zahl;
...

/* Read number from keyboard */
printf("Eine Zahl bitte: ");
scanf("%d", &zahl);
```

- Schreiben Sie ein Programm, das eine vom Benutzer eingegebene Zahl  $n \geq 0$  einliest, und anschließend die Zahlen  $1 \dots n$  am Bildschirm ausgibt.
- Ändern Sie ihr Programm so ab, so dass nun die Fakultät von  $n$  berechnet und am Ende ausgegeben wird.  
(ACHTUNG: Was passiert, wenn  $n$  zu groß gewählt wurde, und wieso passiert dies? Testen Sie z.B.  $n = 20$ .)
- Ändern Sie ihr Programm nun so ab, dass es vom Benutzer zwei Zahlen  $n$  und  $k$  verlangt, und anschließend  $\binom{n}{k}$  berechnet und ausgibt.
  - Benutzen Sie dazu zunächst die Methode, um die Fakultät zu berechnen, die Sie schon implementiert haben.

- ii. (Insbesondere für Fortgeschrittene)

Überlegen Sie sich eine geschickte Implementierung, die es erlaubt, die Zahlen  $n$  und  $k$  so groß wie möglich zu wählen, und immer noch das korrekte Ergebnis zu erhalten.

### Aufgabe C13 (Gerade oder ungerade?)

Schreiben Sie ein Programm, das eine Zahl einliest und für diese Zahl ausgibt, ob sie

- (a) positiv oder negativ bzw. ob sie
- (b) gerade oder ungerade

ist. Dieser Schritt soll so lange wiederholt werden, bis 0 eingegeben wird.

**Hinweis:** Um zu testen, ob eine Zahl gerade oder ungerade ist, kann man sich einen der beiden folgenden Operatoren zunutze machen (probieren Sie ruhig mal beide aus):

% Der %-Operator liefert den Rest einer ganzzahligen Division, d.h.  $5 \% 3$  liefert 2.

/ Dies ist der Divisionsoperator. Sind beide Argumente des Operators ganzzahlig, dann wird *ganzzahlige Division* durchgeführt. Dies entspricht einer Division mit Rest, bei der der Rest ignoriert wird. Demnach liefert  $5 / 3$  den Wert 1.

### Aufgabe C14 (Prä- und Postinkrement)

Betrachten Sie das folgende Programm

```
#include <stdio.h>

int main (void)
{
    int x = 5;

    printf("%d, %d\n", ++x,x--);

    return 0;
}
```

Was für eine Ausgabe erwarten Sie? Was wird tatsächlich ausgegeben?

**Hinweis:** Aufgrund dieses Ergebnisses, sollte man möglichst nicht mehrere Prä- und Postinkrement-Operatoren in einem Statement kombinieren, erst recht nicht für ein und dieselbe Variable.

### Aufgabe C15 (Schleifen, die zweite)

Grundsätzlich kann man jede **for**-Schleife auch als **while**-Schleife schreiben und umgekehrt.

- (a) Was berechnet das folgende Code-Fragment ( $x$  und  $n$  sind vom Typ `int`)?

```
x = -1;
while (n > 0){
    n = n / 3;
    x++;
}
```

Schreiben Sie die **while**-Schleife als **for**-Schleife.

- (b) Wann denken Sie ist eine **for**-Schleife, wann eine **while**-Schleife passender?