



## Einführung in die Informatik II

Prof. Dr. Andrey Rybalchenko, M.Sc. Ruslán Ledesma Garza

Dieses Blatt behandelt Kapitel 1.1 - 1.9 aus dem Buch zur Vorlesung!

**Aufgabe 1.3 (Signum)** Schreiben Sie ein Prozedur `signum : int → int`, die für negative Argumente `-1`, für positive Argumente `1`, und `0` für das Ergebnis `0` liefert.

**Aufgabe 1.4** Schreiben Sie ein Prozedur `hoch17 : int → int`, die zu einer Zahl  $x$  die Potenz  $x^{17}$  berechnet. Dabei sollen möglichst wenige Multiplikationen verwendet werden. Schreiben Sie die Prozedur auf zwei Arten: Mit einer Hilfsprozedur und mit lokalen Deklarationen.

### Aufgabe 1.5

- Geben Sie ein Tupel mit 3 Positionen und nur einer Komponente an.
- Geben Sie einen Tupelausdruck an, der den Typ `int * (bool * (int * unit))` hat.
- Geben Sie ein Paar an, dessen erste Komponente ein Paar und dessen zweite Komponente ein Tripel ist.

**Aufgabe 1.7** Schreiben Sie eine Prozedur `max : int * int * int → int`, die zu drei Zahlen die größte liefert, auf zwei Arten:

- Benutzen Sie keine Hilfsprozedur und drei Konditionale.
- Benutzen Sie eine Hilfsprozedur und insgesamt nur ein Konditional.

**Aufgabe 1.10** Schreiben Sie eine Prozedur `teilbar : int * int → bool`, die für  $(x, y)$  testet, ob  $x$  durch  $y$  ohne Rest teilbar ist.

**Aufgabe 1.11** (Zeitangaben) Oft gibt man eine Zeitdauer im HMS-Format mit Stunden, Minuten und Sekunden an. Beispielsweise ist `2h5m26s` eine hervorragende Zeit für einen Marathonlauf.

- Schreiben Sie eine Prozedur `sec : int * int * int → int`, die vom HMS-Format in Sekunden umrechnet. Beispielsweise soll `sec(1,1,6)` die Zahl `3666` liefern.
- Schreiben Sie eine Prozedur `hms : int → int * int * int`, die eine in Sekunden angegebene Zeit in das HMS-Format umrechnet. Beispielsweise soll `hms 3666` das Tupel `(1,1,6)` liefern. Berechnen Sie die Komponenten des Tupels mithilfe lokaler Deklarationen.

**Aufgabe 1.13** Schreiben Sie eine rekursive Prozedur `mul : int * int → int`, die das Produkt einer natürlichen und einer ganzen Zahl durch wiederholte Addition berechnet. Beschreiben Sie den zugrunde liegenden Algorithmus zunächst mit Rekursionsgleichungen.

**Aufgabe 1.14** Der ganzzahlige Quotient `x div y` lässt sich aus  $x$  durch wiederholtes Subtrahieren von  $y$  bestimmen. Schreiben Sie eine rekursive Prozedur `mydiv : int * int → int`, die für  $x \geq 0$  und  $y \geq 1$  das Ergebnis `x div y` liefert. Geben Sie zunächst Rekursionsgleichungen für `x div y` an.

**Aufgabe 1.15** Auch der ganzzahlige Rest  $x \bmod y$  lässt sich aus  $x$  durch wiederholtes Subtrahieren von  $y$  bestimmen. Schreiben Sie eine rekursive Prozedur  $\text{mymod} : \text{int} * \text{int} \rightarrow \text{int}$ , die für  $x \geq 0$  und  $y \geq 1$  das Ergebnis  $x \bmod y$  liefert. Geben Sie dazu zunächst Rekursionsgleichungen für  $x \bmod y$  an.

**Aufgabe 1.15 (Stelligkeit)** Schreiben Sie eine rekursive Prozedur  $\text{stell} : \text{int} \rightarrow \text{int}$ , die zu einer Zahl die Stelligkeit ihrer Dezimaldarstellung liefert. Beispielsweise soll  $\text{stell } 117 = 3$  gelten. Geben Sie zunächst die Rekursionsgleichungen für  $\text{stell}$  an. Verwenden Sie ganzzahlige Division durch 10, um die Zahl zu erhalten, die durch Streichen der letzten Ziffer entsteht.

**Aufgabe 1.17 (Quersumme)** Schreiben Sie eine rekursive Prozedur  $\text{quer} : \text{int} \rightarrow \text{int}$ , die die Quersumme einer ganzen Zahl berechnet. Die Quersumme einer Zahl ist die Summe ihrer Dezimalziffern. Beispielsweise hat die Zahl  $-3754$  die Quersumme 19. Geben Sie zunächst die Rekursionsgleichungen für  $\text{quer}$  an. Verwenden Sie Restbestimmung modulo 10, um die letzte Ziffer einer Zahl zu bestimmen.