



Hinweis: In einigen PDF-Programmen kann man mit „ALT + markieren“ Spaltenweise kopieren!

1 Kontrollstrukturen

- (i) In der folgenden Aufgabe sollst Du das T_EX-Register `\month` nutzen. Abhängig vom Monat gibt es die Zahlen 1 – 12 aus. Gib den Inhalt des Registers aus.
- (ii) Nutze eine Kontrollstruktur, um statt der Zahlen, die Namen der Monate auszugeben.
- (iii) Dein Kommilitone hat folgende (sehr einfache) Modulusfunktion geschrieben, hat jedoch keine Möglichkeit sie zu testen. Er bittet Dich, die Funktion zu testen und eventuelle Fehler zu beseitigen:

```

1  % Funktion berechnet den Modulus
2  % Aufruf mit: \mod{Zahl}{Modulus}
3
4  \newcommand{\mod}[2]{
5  \newcounter{val}
6  \newcounter{mod}
7  \setcounter{val}{#1}
8  \setcounter{mod}{#2}
9
10 \loop\ifnum\value{val}>0
11 \advance \value{val} by -\value{mod}
12 \repeat
13 \loop\ifnum\value{val}=0
14 \advance\value{val} by \value{mod}
15 \repeat
16 \theval
17 }

```

Teste verschiedene Werte, z. B. $\{9\}\{3\} \stackrel{?}{=} 0$, $\{-19\}\{13\} \stackrel{?}{=} 7$, $\{33123\}\{113\} \stackrel{?}{=} 14$.

- (iv) Nachdem Du die Funktion erfolgreich korrigiert hast, sollst Du nun einen Schleife schreiben, die alle Zahlen, welche durch 3 teilbar sind aufzählt (bspw. bis 100).
- (v) Stellt es ein Problem dar, wenn Du nun eine weitere Funktion schreibst, die einen gleichnamigen Zähler nutzt, wie er bereits in der Modulusfunktion genutzt wurde?

2 Makrodefinition und Expansion

- (i) Schreibe ein Makro, welches beim Aufruf über `\Ankunft{10}{27}` folgendes ausgibt: „Die Ankunft ist um 10:27 Uhr“.
- (ii) Ein Wichtiges Konzept in L^AT_EX ist die Expansion. Schau Dir dazu zunächst folgende Seite an: `\expandafter`. Hier findest Du ebenfalls die Definition von `\edef`. Nun solltest nun in der Lage sein, folgenden Sequenzen zu verstehen:



```

1 % \def ist das TeX Pendant zu \newcommand in LaTeX
2 \def\xx[#1]{...}
3 \def\yy{[ABC]}
4 \def\zz{\yy}
5 \edef\temp
6
7 % 1. Fall:
8 \xx\yy
9 % 2. Fall:
10 \expandafter\xx\yy
11 % 3. Fall:
12 \expandafter\xx\zz
13 % 4. Fall:
14 \edef\temp{\noexpand\xx\zz}

```

- (iii) Was sind zerbrechliche Befehle? Mit welchem Befehl kannst Du zerbrechliche Befehle robust machen? Was hat das mit Expansion zu tun?

3 Komplexere Quellcodes verstehen

Hier mal eine der „typischen“ Hürden aus dem Buch „L^AT_EX-Hacks“. Das hier verwendete Grunderüst wird gleich mehrfach im Buch verwendet, daher ist es besonders schade, dass es in keinem der Hacks einmal zusammenhängend und komplett lauffähig vorgestellt wird! Bei diesem Hack geht es darum, Musterlösungen passend zu den Aufgaben zu erstellen, in etwa so:

```

1 \begin{exercise}
2 Was ist das Lieblingsobst eines Unix-Administrators?
3 \begin{solution}
4 Die Grepfruit.
5 \end{solution}
6 \end{exercise}

```

Der Nutzen davon ist, dass man passender Stelle nun durch `\includesolutions` die Musterlösungen einbinden kann, die passend nummeriert zu den Aufgaben sind. Weiterhin kann man mit dieser Art von Grunderüst viele weitere interessante Dinge ermöglichen (Shell-Kommandos aus L^AT_EX heraus ausführen, GnuPlot einbetten, Trainer- und Teilnehmerversionen von Dokumenten erstellen, ...).

Alle relevanten Codeabschnitte die man dem Buch entnehmen kann sind:

```

1 % Datei zum Schreiben oeffnen:
2 \newwrite\ex@file
3 \immediate\openout\ex@file=\jobname.sol
4
5 % Wird bei \begin{solution} ausgefuehrt:
6 \newcommand*\@bsphack
7 \let\do\@makeother\dospecials

```



```
8 \catcode'\^^M\active
9 \def\verbatim@processline{%
10 \immediate\write\ex@file{\the\verbatim@line}}%
11 \immediate\write\ex@file{%
12 \string\begin{ex@solution}{\theexercise}}% <--- wichtig?
13 \verbatim@start}
14
15 % Wird bei \end{solution} ausgeführt:
16 \def\endsolution{%
17 \immediate\write\ex@file{%
18 \string\end{ex@solution}}\@esphack}
19
20 % Definiert die solution-Umgebung
21 \newenvironment{ex@solution}[1]{\paragraph{#1}}{}
22
23 % Befehl zum Einbinden der Musterloesungen
24 \newcommand*\includesolutions{%
25 \immediate\closeout\ex@file
26 \section{Musterloesungen}
27 \InputIfFileExists{\jobname.sol}{}{}}
```

Bitte beachte, dass Du das Paket `verbatim` einbindest. Insgesamt sind **5** weitere Befehle nötig, damit das Beispiel wie gewünscht funktioniert!

Tipp: Die entscheidende Zeile ist markiert ;-)