

Aufgabe 1

Geben Sie an, wie sich die folgenden Bedingungen mit Hilfe von `test` in einer `if ... then`-Fallunterscheidung formulieren lassen. Der erste Eintrag dient als Beispiel. In der Bedingung enthaltene Buchstaben a,b,c stehen für Variablen.

	Bedingung	test
Bsp.	<code>a=3</code>	<code>if test \$a -eq 3</code>
a)	<code>b ≥ 17</code>	
b)	<code>b ≠ "demo"</code>	
c)	a ist ein Verzeichnis	
d)	Das Skript hat 3 Aufrufparameter	
e)	<code>a > 0</code> oder <code>b = 2</code>	

Aufgabe 2

Im Jahr 2017 entdeckten aufmerksame Softwareentwickler, dass ein Aufruf von `man` ohne Parameter, der genau um 00:30 (also zwischen 00:30:00 und 00:30:59) erfolgt, nicht wie erwartet ausgibt "What manual page do you want?" sondern "gimme gimme gimme...". Siehe

<https://www.heise.de/developer/meldung/gimme-gimme-gimme-Automatischer-Test-stolpert-ueber-Easter-Egg-im-man-Tool-3898570.html>

Schreiben Sie ein Skript `man`, das — falls es um 00:30 ohne weitere Parameter aufgerufen wird — "gimme gimme gimme..." ausgibt, und sich ansonsten genau wie der normale `man`-Befehl verhält. (Hinweis: erinnern Sie sich an `date`. Dann kann mittels `grep` und `if` alles erledigt werden. Zum Testen können Sie eine andere Uhrzeit benutzen, z.B. 16:??)

Aufgabe 3

- Geben Sie einen Kommandozeilenausdruck (= eine Eingabe) an, um den Wert von (3 mal 7 minus 5) hoch 3 auszurechnen.
- Geben Sie die Kommandozeilenausdrücke an, um der Variablen `y` den Wert 8 zuzuweisen, anschließend den Wert von `y+2` an die Variable `x` zuzuweisen, und zum Schluss den Wert von `x` auszugeben.

Aufgabe 4

Unter der URL

`http://www.math.uni-bielefeld.de/~frettlöe/teach/unix/pfade.txt`
können Sie eine Datei mit dem folgenden Inhalt herunterladen:

```
Bielefeld->Gütersloh->Osnabrück->Münster
Gütersloh->Rheda-Wiedenbrück->Warendorf
Münster->Warendorf->Bielefeld->Bad Salzuflen
...
```

Schreiben Sie ein Skript, das für eine anzugebende Stadt ermittelt, wie oft sie als Start- und Zielpunkt eines Pfades vorkommt. Geben Sie außerdem die Summe der beiden Werte aus. Beispielaufufe:

```
$ ./suche-nach.sh Bielefeld
3 Pfade beginnen in Bielefeld.
7 Pfade enden in Bielefeld.
10 Pfade insgesamt mit Bielefeld als Start oder Ziel.
$ ./suche-nach.sh Münster
3 Pfade beginnen in Münster.
3 Pfade enden in Münster.
6 Pfade insgesamt mit Münster als Start oder Ziel.
```

Lösungshinweise: Zur Lösung der Aufgabe benötigen Sie das Ablegen von Werten in Variablen, das Rechnen mit Variablen sowie die Ausgabe von Variablen mit dem `echo`-Befehl. Um innerhalb der Pfad-Datei nach Start- und Zielorten zu suchen, sehen Sie sich noch einmal Vorlesung 4, Folie 34, zur Anwendung von `grep` an. In dieser Aufgabe werden keine Fallunterscheidungen benötigt.

Aufgabe 5

Schreiben Sie ein Skript `mkdir2.sh`, das beim Aufruf einen Unterverzeichnisnamen erhält und damit Folgendes macht:

- Falls beim Aufruf kein Verzeichnisname angegeben wurde, wird eine Hilfestellung zur Benutzung ausgegeben.
- Falls sich im aktuellen Verzeichnis bereits eine Datei oder ein Unterverzeichnis mit diesem Namen befinden, wird eine entsprechende Meldung ausgegeben.
- Anderenfalls wird ein leeres Unterverzeichnis mit dem angegebenen Namen angelegt.

Beispielaufufe:

```
$ ls
briefe text.txt
$ ./mkdir2.sh
Aufruf: ./mkdir2.sh unterverzeichnis
$ ./mkdir2.sh briefe
briefe existiert schon als Verzeichnis!
$ ./mkdir2.sh text.txt
text.txt existiert schon als Datei!
$ ./mkdir2.sh ablage
Verzeichnis ablage wurde angelegt.
```

Zusatzaufgabe

Schreiben Sie ein Shellskript `rechner.sh` das die vier Grundrechenarten (+,-,x,/) auf zwei Zahlen anwenden kann. Verwenden Sie den Kleinbuchstaben "x" für das Multiplikationssymbol und lösen Sie die Aufgabe mit Hilfe von `case` (siehe Folien Vorlesung 8). Sehen Sie eine Fehlerbehandlung für falsche Aufrufmöglichkeiten vor.

Beispielaufrufe:

```
$ ./rechner.sh 5 + 7
12
$ ./rechner.sh 40 - 12
28
$ ./rechner.sh 8 x 9
72
$ ./rechner.sh 64 / 16
4
$ ./rechner.sh 64 y 16
Unbekannter Operator: y
$ ./rechner.sh
Aufruf: rechner.sh a [+ - x /] b
```

Downloads (Folien, Übungsblätter)

<https://www.math.uni-bielefeld.de/~frettloe/teach/unix22.html>

Hinweise zu den Übungen

Die Übungen dienen dem Erlernen von Linux. Es gibt keine Abgabepflicht, es gibt überhaupt keine Abgaben. In der **A&D-Klausur** am Ende des Semesters werden allerdings Linux-Fragen vorkommen. Außerdem brauchen Sie im Verlaufe des Studiums solide Linuxkenntnisse. Daher ist es sinnvoll, dass Sie die Übungen entweder selbständig lösen, oder aber eines der Tutorien besuchen und die Übungen dort bearbeiten. Darüber hinaus können Sie in den Tutorien den Tutoren Fragen stellen zu Übungen und Vorlesung.