

Vorsemesterkurs Informatik

Sommersemester 2019

Aufgabenblatt Nr. 1B

Aufgabe 1 (Verzeichnisse)

- Finden Sie heraus, wie Ihr Homeverzeichnis auf Ihrem Rechner, Ihr Username und der Name des Rechners, auf dem Sie arbeiten, heißen.
- Legen Sie in Ihrem Homeverzeichnis die im folgenden Baum dargestellten Unterverzeichnisse ausschließlich unter Verwendung der Shell-Befehle `cd` und `mkdir` an, ohne dabei ein `/` zu benutzen.

```
vorkurs
|-- verzeichnis-1
|   |-- unterverzeichnis-1-1
|   |-- unterverzeichnis-1-2
|   '-- unterverzeichnis-1-3
|-- verzeichnis-2
|   '-- .verstecktes_unterverzeichnis-2-1
'-- verzeichnis-3
    |-- unterverzeichnis-3-1
    '-- unterverzeichnis-3-2
```

Welche Kommandos haben Sie eingegeben?

- Wechseln Sie in Ihr Homeverzeichnis und führen Sie den Befehl `ls` mit den unten genannten Optionen (auch mit allen Kombinationen) und dem Parameter `vorkurs` aus. Erläutern Sie die Effekte der unterschiedlichen Optionen (schauen Sie dazu in die Man Page von `ls` (mittels `man ls`)).
 - `-a`
 - `-l`
 - `-R`
- Informieren Sie sich über die Verwendung des Befehls `rm`, indem Sie die Man Page dazu durchlesen (mittels `man rm`) und/oder im Internet danach suchen. Löschen Sie die angelegten Unterverzeichnisse von `vorkurs` mithilfe des Befehls `rm`.
- Erstellen Sie Unterverzeichnisse von `vorkurs` erneut jedoch unter Verwendung des Kommandos `mkdirhier` (studieren Sie vorher die dazugehörige Man Page). Löschen Sie anschließend die Unterverzeichnisse von `vorkurs` erneut.

Aufgabe 2 (Textdateien)

a) Legen Sie mithilfe eines Editors eine Datei namens `IrgendeinText.txt` im Verzeichnis `vorkurs` an, und schreiben Sie irgendeinen Text der mindestens 100 Zeilen besitzt und das Wort "Informatik" enthält in die Datei. Sichern Sie die Datei.

b) Führen Sie das Kommando

```
tail -n 20 IrgendeinText.txt | head > IET.txt
```

aus und finden Sie heraus, was dieses Kommando anstellt. Hinweis: Lesen Sie z.B. die Webseite http://www.selflinux.de/selflinux/html/bash_basic03.html.

c) Verwenden Sie das Kommando `mv`, um die Datei `IrgendeinText.txt` umzubenennen in `MeinText.txt`

d) Verwenden Sie das Kommando `cp`, um eine Kopie der Datei `MeinText.txt` namens `NochmalMeinText.txt` zu erstellen

e) Wechseln Sie in Ihr Homeverzeichnis und erstellen Sie eine Kopie des Verzeichnisses `vorkurs` namens `vorkurs2`. Welche Option benötigt `cp` hierfür?

f) Benennen Sie das Verzeichnis `vorkurs2` in `vorkurs-2` um.

g) Führen Sie im Homeverzeichnis das Kommando

```
grep -R Informatik *
```

aus. Welche Ausgabe erhalten Sie? Finden Sie anhand der Man Page zu `grep` heraus, was das Kommando macht.

Aufgabe 3 (Haskell Interpreter: GHCi)

Starten Sie den Haskell Interpreter `GHCi` aus Ihrem Homeverzeichnis.

a) Verschaffen Sie sich einen Überblick über die Bedienung des `GHCi`, indem Sie sich die Hilfe im `GHCi` anzeigen lassen.

b) Lassen Sie sich im `GHCi` das Verzeichnis anzeigen, in dem Sie sich befinden. Das Kommando `:!` ist hierbei hilfreich. Wechseln Sie anschließend in das Verzeichnis `~/vorkurs` ohne den `GHCi` zu verlassen.

c) Geben Sie zu jedem der folgenden arithmetischen Ausdrücke den entsprechenden Haskell-Ausdruck an und lassen Sie den `GHCi` jeweils dessen Wert berechnen. Füllen Sie dabei die folgende Tabelle aus.

Arithmetischer Ausdruck	Ausdruck in Haskell	Ergebnis im GHCi
$1 + 3 + 5 + 7 + 9$		
$(15 - 6) \cdot 3 + 12 \cdot 2$		
$\frac{1}{2} + \frac{1}{4}$		
$(\frac{1}{3} + \frac{1}{2}) + \frac{1}{6}$		
$\frac{1}{3} + (\frac{1}{2} + \frac{1}{6})$		
$\frac{1}{0}$		
$-1 \cdot 2$		
$2 \cdot -1$		
2^{2^2}		

Hinweis: Verwenden Sie die Operatoren `*`, `/` und `^` zur Multiplikation, Division und Potenzierung.

Aufgabe 4 (Funktionalität testen)

Auf der Webseite zum Vorkurs (<http://vorkurs.informatik.uni-frankfurt.de/>) finden Sie eine Datei `magic.hs`. Laden Sie diese herunter und laden Sie sie anschließend in den GHCi. Die Datei stellt die Funktionen `fun1`, `fun2`, `fun3`, `fun4` und `fun5` bereit. Diese erwarten eine Zeichenkette als Eingabe und liefern eine veränderte Zeichenkette. Ein Test ist z.B. `fun1 "Hallo"`.

Finden Sie durch *Testen* der Funktionen `fun1` bis `fun5` heraus, welche der folgenden Änderungen diese Funktionen auf Zeichenketten durchführen (Mehrfachantworten pro Funktion sind möglich) und kreuzen Sie die entsprechenden Antworten an. Beachten Sie, dass der Quellcode der Datei `magic.hs` mit Absicht nahezu unverständlich ist.

Wirkung	fun1	fun2	fun3	fun4	fun5
1) macht aus Kleinbuchstaben Grossbuchstaben	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
2) macht aus Grossbuchstaben Kleinbuchstaben	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
3) macht aus dem i-ten Buchstaben des Alphabets die Zahl i	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
4) erniedrigt alle Ziffern (ausser der 0) um 1	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
5) macht aus jedem Fragezeichen ein Ausrufezeichen	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
6) ersetzt alle Ziffern (außer 0), durch ihre Darstellung als römische Zahl	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
7) entfernt alle runden Klammern	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
8) verdoppelt alle Vorkommen der Buchstaben x,y,z,X,Y,Z	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
9) vertauscht in Sätzen manche Worte	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
10) entfernt Worte, die mehrfach im Text auftauchen	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
11) löscht Leerzeichen, falls Worte mit mehr als einem Leerzeichen getrennt sind	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein	<input type="checkbox"/> Ja <input type="checkbox"/> Nein

Aufgabe 5 (Boolesche Ausdrücke und Rätsel lösen)

a) Überlegen Sie sich für die folgenden Booleschen Ausdrücke, *welchen* Wahrheitswert diese jeweils darstellen. Überprüfen Sie *anschließend* Ihr Ergebnis, indem Sie die Ausdrücke im GHCi auswerten lassen. Hierfür müssen Sie die Ausdrücke in Haskell-Notation überführen (d.h. `True` und `False` für die Wahrheitswerte und `not`, `&&`, `||` für \neg , \wedge , \vee verwenden).

- \neg wahr
- wahr \vee falsch
- $\neg((\text{falsch} \vee \text{wahr}) \wedge (\neg\text{falsch}))$
- $\neg((\text{falsch} \vee (\neg\text{falsch})) \wedge (\neg\text{wahr}))$
- $\neg(\neg((\neg\text{wahr}) \vee \text{falsch}))$

b) Lügenbolde lügen immer, während Wahrsager immer die Wahrheit sagen. Jeder der drei Brüder Knasi, Knesi und Knosi ist entweder ein Lügenbold oder ein Wahrsager. Knasi sagt: “Wir Brüder sind alle Lügenbolde”. Knesi sagt: “Genauer einer von uns sagt die Wahrheit”.

Formalisieren Sie das Rätsel mit Aussagenlogik in Haskell, indem Sie zunächst definieren

```
knasi_ist_wahrsager = undefined
knesi_ist_wahrsager = undefined
knosi_ist_wahrsager = undefined
```

und anschließend die beiden Aussagen in Haskell formulieren und verknüpfen:

```
knasis_aussage = ...
knesis_aussage = ...

beide_aussagen = ...
```

Probieren Sie anschließend durch alle Belegungen (`True` oder `False`) für die Variablen `knasi_ist_wahrsager`, `knesi_ist_wahrsager`, `knosi_ist_wahrsager` aus (indem Sie `undefined` durch die Wahrheitswerte ersetzen) und lassen Sie den Wert von `beide_aussagen` berechnen.

Für welche Belegung(en) sind beide Aussagen wahr, d.h. wer ist Lügenbold und wer ist Wahrsager?