

# Wochenauflagen

## Aufgabe 01

Eine Liste von Personen mit ihrer Rolle (admin oder stud) aufgelistet und ob sie ein Notebook haben.

(Die dritte Spalte (a oder b) ist zunächst bedeutungslos.)

Liste alle Datensätze von Personen sortiert nach ihrem Namen auf,

bei denen in der 2. Spalte stud steht und die kein Notebook haben!

Kopiere das Ergebnis in die Datei /home/step2022/wochenblogs/DEINACCOUNT/loesung-01.txt

wobei DEINACCOUNT für den Namen steht, mit dem Du

Dich bei hopper anmeldest. Die Datei muss genau so heißen!

Prüfe die Richtigkeit Deiner Lösung mit dem Kommando

step-pruefe-aufgabe-01.sh

```
1 #!/bin/bash
2 cat user.txt | grep "stud" | grep "nein" > /home/step2022/wochenblogs/mohalzubaidy/loesung-01.txt
3
```

## Aufgabe 02

Wochenauflage 02 :

In /home/step2022/data/person.txt findet Ihr wieder eine Liste von Personen. Dieses Mal sind Vorname,

Nachname, Geburtsjahr und Rolle angegeben.

Schreibt ein Programm (ein Skript), das nur die

Datensätze heraus sucht, bei denen

der Vorname entweder oliver oder karin ist,

oder

die Rolle stud ist und das Geburtsjahr vor 2000 liegt

oder

der Nachname mit einem f beginnt und der Vorname

mit einem f beginnt.

Das Ergebnis soll sortiert (ohne Angabe eines keys)

und mit einer Zeilennummer ohne führende Leerzeichen

ausgegeben werden.

```
1 vorname=fabian nachname=falke geboren=1991 rollestud
2 vorname=fabian nachname=hefleige geboren=1996 rollestud
...
```

In der obigen Bedingungsformulierung lest Aussagen auf einer Zeile als geklammerte Aussagen,

Kopiert das Ergebnis nach

```
/home/step2022/wochenblogs/DEINACCOUNT/loesung-02.txt
```

und prüft das Ergebnis mit dem Befehl

step-pruefe-aufgabe-02.sh

```
1 #!/bin/bash
2 cat person.txt | grep "oliver" | grep "stud" | sort -t"," -k 1,2 > mmsi-name.txt
3 cat person.txt | grep "karin" | grep "stud" | sort -t"," -k 1,2 > mmsi-name.txt
4 cat person.txt | grep "f" | grep "f" | sort -t"," -k 1,2 > mmsi-name.txt
5 cat person.txt | grep "hefleige" | sort -t"," -k 1,2 > mmsi-name.txt
6 cat person.txt | grep "fabian" | sort -t"," -k 1,2 > mmsi-name.txt
7 cat person.txt | grep "luebeck" | sort -t"," -k 1,2 > mmsi-name.txt
8 cat person.txt | grep "hamburg" | sort -t"," -k 1,2 > mmsi-name.txt
9 cat person.txt | grep "f" | sort -t"," -k 1,2 > mmsi-name.txt
10 cp loesung-02.txt /home/step2022/wochenblogs/losakbari/
```

## Aufgabe 03

Teil 1: filtert aus der Datei /home/step2022/data/rhodes.txt die MMSI (3. Spalte) und den Namen (6. Spalte, wenn in der 2. Spalte die Zahl 5 steht). Sortiert das Ergebnis und sorgt dafür, dass jede MMSI nur einmal in dem Ergebnis ist.

Schreibt das Ergebnis in eine Datei mit dem Namen mmsi-name.txt

Teil 2: Lest Euch die man-Page des Programmes join durch.

Versucht, zu verstehen, was das Programm kann. Wenn die man-Page nicht genügt, könnt ihr natürlich auch im Internet nach Informationen suchen. Zu Eurem Leben in der Informatik wird es gehören, sich neue Informationen schnell anzusegnen

und dafür in Dokumentationen zu lesen.

Gute Informationen - auch zu anderen Programmen unter Linux -

gibt es beispielweise dort:

<https://www.geeksforgeeks.org/join-command-linux/>

Teil 3: In /home/step2022/data/mmsi-harbor.txt findet sich eine Liste von aktiven Hafenbesuch von Schiffen.

Findet die Namen von Schiffen, die sowohl in Luebeck als

auch in Hamburg waren. Sortiert die Liste und kopiert sie unter dem Namen loesung-03.txt

In Euer wochenlog-Verzeichnis in /home/step2022/wochenblogs/

Prüft Eure Lösung mit step-pruefe-aufgabe-03.sh

```
1 #!/bin/bash
2 cat rhodes.txt | grep "^\[|]*\$" | cut -d " " -f 3,6 | sort | uniq > mmsi-name.txt
3
4 cat mmsi-harbor.txt | grep "luebeck" | cut -d " " -f 1 | sort > luebeck.txt
5 cat mmsi-harbor.txt | grep "hamburg" | cut -d " " -f 1 | sort > hamburg.txt
6
7 join luebeck.txt hamburg.txt > luebeck-hamburg.txt
8 join -t " " mmsi-name.txt luebeck-hamburg.txt > loesung-03.txt
9
10 cp loesung-03.txt /home/step2022/wochenblogs/losakbari/
11 step-pruefe-aufgabe-03.sh
```

## Aufgabe 04

Erzeugt eine korrekte (mit tidy geprüfte)

Html-Datei mit dem Namen aufgabe-04.html

in Eurem Wochenblogverzeichnis.

In der Datei soll es in dem body

eine Tabelle mit vier Spalten geben:

Nr|MMSI|Name|Hafen

Die erste Spalte enthält eine fortlaufende

Nummer, die bei 1 beginnt.

Die Spalten zwei und drei enthalten die

MMSI und den Namen von den Schiffen, die

in der Datei /home/step2022/data/rhodes.txt

vorkommen, wenn sie laut

/home/step2022/data/mmsi-harbor.txt

wenigstens in einem Hafen gewesen sind.

Die Ergebnistabelle soll nach den

Namen der Schiffe sortiert sein.

Ein Schiff kann also in mehreren Häfen

angelegt hat.

Die erste Zeile der Tabelle soll die

Spaltennamen wie oben angegeben enthalten - wird aber nicht als Zeile

1 gezählt.

Prüft das Ergebnis wieder mit

step-pruefe-aufgabe-04.sh

```
1 #!/bin/bash
2 cat rhodes.txt | grep "^\[|]*\$" | cut -d " " -f 1,2,6 | sort -t"," -k 2 | uniq -20 > mmsi-name.txt
3
4 cat mmsi-harbor.txt | while IFS='|' read name mmsi
5 do
6 echo "$name|$mmsi" > /home/step2022/wochenblogs/losakbari/loesung-04.txt
7 done
8
9 cat mmsi-name.txt > schritt-03.txt
10 echo "<!DOCTYPE html><html>
11 <head>
12 <title>Das ist der Titel</title>
13 </head>
14 <body>
15 <table>
16 <tr>
17 <td>Name</td>
18 <td>MMSI</td>
19 <td>Hafen</td>
20 <td>Schiff</td>
21 </tr>
22 <tr>
23 <td>Name</td>
24 <td>MMSI</td>
25 <td>Hafen</td>
26 <td>Schiff</td>
27 </tr>
28 <tr>
29 <td>Name</td>
30 <td>MMSI</td>
31 <td>Hafen</td>
32 <td>Schiff</td>
33 </tr>
34 </table>
35 </body>
36 </html>" > schritt-04.txt
37 tidy -q schritt-04.txt > schritt-04.html
38 cp aufgabe-04.html /home/step2022/wochenblogs/losakbari/
39 step-pruefe-aufgabe-04.sh
```

## Aufgabe 05

Erstellt in Eurem Wochenblog-Verzeichnis ein Verzeichnis mit dem Namen loesung-05

Erstellt mit den neu erlernten Konzepten ein Programm, das in diesem Verzeichnis für jede MMSI,

die in der Datei /home/step2022/data/rhodes.txt

existiert, eine Datei anlegt, deren Name nur aus der

MMSI

(ohne Endung) besteht. Erstellt die Datei genau dann,

wenn sie noch nicht existiert und ein Name für die

MMSI bekannt ist.

Schreibt für jede MMSI in die erste Zeile der Datei

nameNAME-DES-SCHIFFES

Darunter schreibt noch einen Eintrag für das erste

Auftauchen der MMSI in einem Typ-5 Eintrag

in der rhodes-Datei nach dem Format:

first:2022-11-02 HH:MM:SS

Am Ende soll es also in dem Verzeichnis loesung-05

eine Datei geben für jede MMSI, die in der

rhodes-Datei existiert und für die

mindestens einen Eintrag vom Typ 5 gibt

(Spalte 2-5).

mit jeweils zwei Zeilen.

Benutzt für die Lösung nur die Elemente, die Ihr

in der MMSI-Lektion gelernt habt. Auch wenn Ihr schon

Lösungen mit anderen Möglichkeiten kennt, sollt

Ihr auch diese hier kennenlernen und ausprobieren.

Prüft die Lösung wieder mit

step-pruefe-aufgabe-05.sh

```
1 #!/bin/bash
2 cat rhodes.txt | grep "^\[|]*\$" | cut -d " " -f 1,2,6 | sort -t"," -k 2 | uniq -20 > mmsi-name.txt
3
4 cat mmsi-harbor.txt | while IFS='|' read name mmsi
5 do
6 echo "$name|$mmsi" > /home/step2022/wochenblogs/losakbari/loesung-05/$mmsi
7 done
8
9 cat mmsi-name.txt > schritt-03.txt
10 echo "<!DOCTYPE html><html>
11 <head>
12 <title>Das ist der Titel</title>
13 </head>
14 <body>
15 <table>
16 <tr>
17 <td>Name</td>
18 <td>MMSI</td>
19 <td>Hafen</td>
20 <td>Schiff</td>
21 </tr>
22 <tr>
23 <td>Name</td>
24 <td>MMSI</td>
25 <td>Hafen</td>
26 <td>Schiff</td>
27 </tr>
28 <tr>
29 <td>Name</td>
30 <td>MMSI</td>
31 <td>Hafen</td>
32 <td>Schiff</td>
33 </tr>
34 </table>
35 </body>
36 </html>" > schritt-04.txt
37 tidy -q schritt-04.txt > schritt-04.html
38 cp aufgabe-05.html /home/step2022/wochenblogs/losakbari/
39 step-pruefe-aufgabe-05.sh
```

## Aufgabe 06

Erstellt für die Datei

/home/step2022/data/teams.txt

eine Verzeichnis, das eine

Datei index.html enthält, in

der eine Liste aller Schiffe als

Tabelle steht (Nr,MMSI,Nam).

Die Tabelle soll sortiert sein

nach dem Namen des Schiffes,

die Spaltennamen in der ersten

Zeile stehen und

wieder beginnend mit 1 nummeriert sein.

Dabei soll in jeder Zeile die MMSI

auch als Html-Anker

<a href="#\$mmsi">\$mmsi</a>

dienen, der auf eine Html-Datei im

selben Verzeichnis verweist,

in der wiederum eine Tabelle

mit Einträgen aller Nachrichten

der Typen 1 und 3 für diese

MMSI steht. Der Inhalt der Tabelle

steht in der Datei teams.txt

und ist in der gleichen Reihenfolge

wie in teams.txt. Der Inhalt der Tabelle

steht in der Datei teams.txt

und ist in der gleichen Reihenfolge

wie in teams.txt. Der Inhalt der Tabelle

steht in der Datei teams.txt

und ist in der gleichen Reihenfolge

wie in teams.txt. Der Inhalt der Tabelle

steht in der Datei teams.txt

und ist in der gleichen Reihenfolge

wie in teams.txt. Der Inhalt der Tabelle

steht in der Datei teams.txt

und ist in der gleichen Reihenfolge

wie in teams.txt. Der Inhalt der Tabelle