

# Wochenaufgaben

## 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 in den Namen steht, mit dem Du Dich bei hopper anmeldest. Die Datei muss genau so heißen! Prüfe die Korrektheit Deiner Lösung mit dem Kommando step-pruefe-aufgabe-01.sh

```
1 |/bin/bash
2 cat user.txt | grep "stud" | grep "nokia" > /home/step2022/wochenblogs/mohatzubaidy/loesung-01.txt
```

## Aufgabe 02

Wochenaufgabe 02 : In /home/step2022/data/person.txt findet Ihr wieder eine Liste mit Personen. Dieses Mal sind Vorname, Nachname, Geburtsjahr und Rolle angegeben. Schreib 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 rolle:stud 2 vorname:fabian nachname:fliege geboren:1996 rolle:stud ... 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 /home/step2022/data/person.txt | grep "oliver" | grep "karin" > dreib.txt
3 grep "stud" person.txt | grep "oliver" | grep "karin" > dreib.txt
4 grep "nokia" person.txt | grep "nokia" > dreib.txt
5 cat dreib.txt | sort | uniq | nl -n 1 -> loesung-02.txt | cp loesung-02.txt /home/step2022/wochenblogs/hosakbari/
6 step-pruefe-aufgabe-02.sh
```

## 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 sortiert dafür, dass jede MMSI nur einmal in dem Ergebnis ist. Schreib 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 anzu eignen und dafür in Dokumentationen zu lesen. Gute Informationen - auch zu anderen Programmen unter Linux - gibt es beispielsweise dort: https://www.geekstorgeeks.org/join-command-linux/ Teil 3: In /home/step2022/data/mmsi-harbor.txt findet sich eine Liste von fiktiven Hafenbesuchen 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 mmsi-harbor.txt | grep "[1-5]" | cut -d " " -f 3,6 | sort | uniq > mmsi-name.txt
3 cat rhodes.txt | grep "5" | cut -d " " -f 1,3,6 | sort -t " " -k 2 | sed "s/ /|/" | nl -n 1 > loesung-03.txt
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 join luebeck.txt hamburg.txt > luebeck-hamburg.txt
7 join -t '|' mmsi-name.txt luebeck-hamburg.txt > loesung-03.txt
8 step-pruefe-aufgabe-03.sh
```

## Aufgabe 04

Erzeugt eine korrekte (mit tidy geprüfte) HTML-Datei mit dem Namen aufgabe-04.html in Eurem Wochenlogverzeichnis. 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 Zeilen erscheinen, wenn es in mehreren Häfen angelegt hat. Die erste Zeile der Tabelle soll die Spaltennamen wie oben angegeben enthalten - wird aber nicht als Zeile gezählt. Prüft das Ergebnis wieder mit step-pruefe-aufgabe-04.sh

```
1 |/bin/bash
2 cat /home/step2022/data/rhodes.txt | grep "[1-5]" | cut -d " " -f 1,3,6 | sort | uniq > schritt-01.txt
3 cat rhodes.txt | grep "5" | cut -d " " -f 1,3,6 | sort -t " " -k 2 | sed "s/ /|/" | nl -n 1 > schritt-02.txt
4 echo "Schritt 02: 05/01/2022"
5 head -n 1 schritt-02.txt | sed "s/|/ /" > schritt-03.txt
6 echo "Schritt 03: 05/01/2022"
7 head -n 1 schritt-03.txt | sed "s/|/ /" > schritt-04.txt
8 echo "Schritt 04: 05/01/2022"
9 echo "Schritt 05: 05/01/2022"
10 echo "Schritt 06: 05/01/2022"
11 sed -n '1,4p' schritt-04.txt | sed "s/|/ /" | sed "s/ /<br />" | sed "s/ /<br />" > schritt-05.txt
12 echo "Schritt 05: 05/01/2022"
13 echo "Schritt 06: 05/01/2022"
14 head -n 1 schritt-05.txt > schritt-06.txt
15 tidy -i schritt-06.txt > aufgabe-04.html
16 cp aufgabe-04.html /home/step2022/wochenblogs/hosakbari
17 step-pruefe-aufgabe-04.sh
```

## Aufgabe 05

Erstellt in Eurem Wochenlog-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. Erstelle die Datei genau dann, wenn sie noch nicht existiert und ein Name für die MMSI bekannt ist. Schreib für jede MMSI in die erste Zeile der Datei name:NAME-DES-SCHIFFES. Darunter schreib 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 es mindestens einen Eintrag vom Typ 5 gibt (Spalte 2-5) mit jeweils zwei Zeilen. Benutzt für die Lösung nur die Elemente, die Ihr in STEP kennengelernt 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 "[1-5]" | cut -d " " -f 1,3,6 | sort -t " " -k 2 | uniq ->2 > mmsi+name.txt
3 cat mmsi+name.txt | while IFS='|' read time msi name
4 do
5 echo "name:$name" > /home/step2022/wochenblogs/hosakbari/loesung-05/$msi
6 done
7 echo "first:$time" >> /home/step2022/wochenblogs/hosakbari/loesung-05/$msi
8 done
9 step-pruefe-aufgabe-05.sh
```

## Aufgabe 06

Erstellt für die Datei /home/step2022/data/rhodes.txt eine Verzeichnis, das eine Datei index.html enthält, in der eine Liste aller Schiffe als Tabelle steht (Nr,MMSI,Name). Die Tabelle soll sortiert sein nach dem Namen des Schiffes, die Spaltennamen in der ersten Zeile aufführen und wieder beginnend mit 1 nummeriert sein. Dabei soll in jeder Zeile die MMSI (<a href="123456.html">123456</a>) dienen, der auf eine HTML-Datei im selben Verzeichnis verweist. In der Verzeichnis eine Tabelle mit Einträgen aller Nachrichten der Typen 1 und 3 für diese MMSI in rhodes.txt sortiert nach dem Zeitstempel steht. In der Tabelle sollen die Spalten Nr,Zeitstempel,MMSI,Longitude,Latitude stehen. (Longitude ist die Spalte, die in Bremerhaven mit 8 beginnt, Latitude ist die Spalte, die mit 53 beginnt - lest die Bedeutung von Longitude und Latitude bitte nach.) Auch hier soll die Tabelle in der ersten Zeile die Spaltennamen enthalten und die Zeilen danach sollen beginnend mit 1 nummeriert sein. Die Namen der MMSI-Dateien sollen nur aus der MMSI gefolgt von .html bestehen. Das gesamte Ergebnis (alle HTML-Dateien) sollen schließlich in Eurem Wochenlog-Verzeichnis in dem Unterverzeichnis loesung-06 stehen. Alle Dateien müssen laut tidy korrektes HTML sein. Außer den Tabellen soll nichts anderes in dem body des Dokumentes stehen. Der Titel des Hauptdokumentes soll Register sein, der der einzelnen MMSI-Dokumente die jeweilige MMSI-Nummer. Während der Arbeit an der Lösung solltet Ihr die Zwischenschritte in Euer Webverzeichnis - ebenfalls unter loesung-06 - kopieren, damit Ihr sie auch jederzeit in Eurem Browser anschauen könnt. Gewöhnt Euch die Tastenkombination zum Neuladen der aktuellen Seite in Eurem Browser an, damit Ihr nicht immer zur Maus greifen müsst. Da hier nun viel und oft in Dateien geschrieben wird, solltet Ihr euch in /tmp/ ein Arbeitsverzeichnis mit Eurem Namen anlegen und darin arbeiten, damit die einzelnen Schritte nicht so lange dauern. Prüft Eure Lösung wieder mit step-pruefe-aufgabe-06.sh

```
1 |/bin/bash
2 cat rhodes.txt | grep "[1-5]" | cut -d " " -f 1,3,6 | sort -t " " -k 2 | uniq ->2 > mmsi+name.txt
3 cat mmsi+name.txt | while IFS='|' read time msi name
4 do
5 echo "name:$name" > /home/step2022/wochenblogs/hosakbari/loesung-05/$msi
6 done
7 echo "first:$time" >> /home/step2022/wochenblogs/hosakbari/loesung-05/$msi
8 done
9 step-pruefe-aufgabe-05.sh
```

## Aufgabe 07

In /home/step2022/data/teams.txt findet sich eine recht lange Liste von virtuellen Teams zusammen mit den Namen der Mitglieder. aaaa:Gislinde Hummel,Loni Spiegel,Janet Unger,Kreszenz Rosenthal,Ina Herrmann aaabingo Wacker,Anneline Frank,Matthias Walner,Irmrud Pfeiffer aacd:Roswita Daumli,Elia Muth,Anton Frenzel,Magda Hornung,Stephan Hering aade:Rosina Winkelmann,Ellen Westphalia,La Sonntag,Lothar Gelger,Susana Rost aaee:Angela Strobel,Karla Dreyer,Julia Maul,Mariola Roos,Jonka Meier aaf:Dorothe Herold,Marica Hagen,Dora Schott ... Erstelle eine Datei mit dem Namen loesung-07.txt in Eurem Wochenlog-Verzeichnis, in der jedes Mitglied einzeln auf einer Zeile zusammen mit dem Team aufgelistet wird. Das erste Wort soll das Team sein, danach der Mitgliedsname, getrennt durch ein Leerzeichen. aaaa Gislinde Hummel aaab Ina Herrmann aaac Janet Unger aaad Kreszenz Rosenthal aaae Loni Spiegel aaab Anneline Muth Die Datei loesung-07.txt soll alphabetisch sortiert sein.

```
1 |/bin/bash
2 sed 's/ /|/g' /home/step2022/data/teams.txt > teams.txt
3 cat teams.txt | while IFS='|' read team a b c d e;do
4 echo "$team $a"
5 done > name1.txt
6 cat teams.txt | while IFS='|' read team a b c d e;do
7 echo "$team $b"
8 done > name2.txt
9 cat teams.txt | while IFS='|' read team a b c d e;do
10 echo "$team $c"
11 done > name3.txt
12 cat teams.txt | while IFS='|' read team a b c d e;do
13 echo "$team $d"
14 done > name4.txt
15 cat teams.txt | while IFS='|' read team a b c d e;do
16 echo "$team $e"
17 done > name5.txt
18 jdt name1.txt > namen.txt
19 cat name2.txt >> namen.txt
20 cat name3.txt >> namen.txt
21 cat name4.txt >> namen.txt
22 cat name5.txt >> namen.txt
23 cat namen.txt | sort | uniq > endlich.txt
24 cat endlich.txt | sed "s/|/|&|/g" | grep "[1-5]" | cut -d " " -f 1 | cut -d " " -f 1 | cut -d " " -f 1 | cut -d " " -f 1
25 cat fertig.txt | sed "s/|/|&|/g" > /home/step2022/wochenblogs/mohatzubaidy/loesung-07.txt
```

## Aufgabe 08

Unter der Web-Adresse https://informatikhs-bremerhaven.de/-step2022/rhodes/ findet sich für jeden Tag der ersten Woche des Dezembers eine Datei der Form 2022-12-dd.txt wobei dd für den Tag steht - also 01, 02, 03, 04 ... Das Format der Dateien entspricht den bekannten Ausgaben von rhodes. Ladet Euch alle 7 Dateien herunter und findet die MMSI und Namen aller Schiffe, die in der Zeit erfasst wurden. Sorgt dafür, dass in Eurem Wochenlog-Verzeichnis in der Datei loesung-08.txt - alle Namen und MMSI - einmalig, klein - mit durchgängig kleinen - geschrieben Namen - sortiert nach dem Namen - Name MMSI - getrennt durch das |-Zeichen stehen. Also: alcdum|211480540 alexander.gogel|211264470 aite.weser|218709000

```
1 |/bin/bash
2 curl https://informatikhs-bremerhaven.de/-step2022/rhodes/2022-12-(01..07).txt > d1.txt
3 cat d1.txt | grep "[1-5]" | cut -d " " -f 3,6 > neuedatei.txt
4 cat neuedatei.txt | tr 'a-z' 'A-Z' > wiederedatei.txt
5 cat wiederedatei.txt | \
6 while IFS='|' read mmsi name; do
7 echo "name:$name"
8 done > nameessi.txt
9 cat nameessi.txt | sort -u > loesung-08.txt
```

## Aufgabe 09

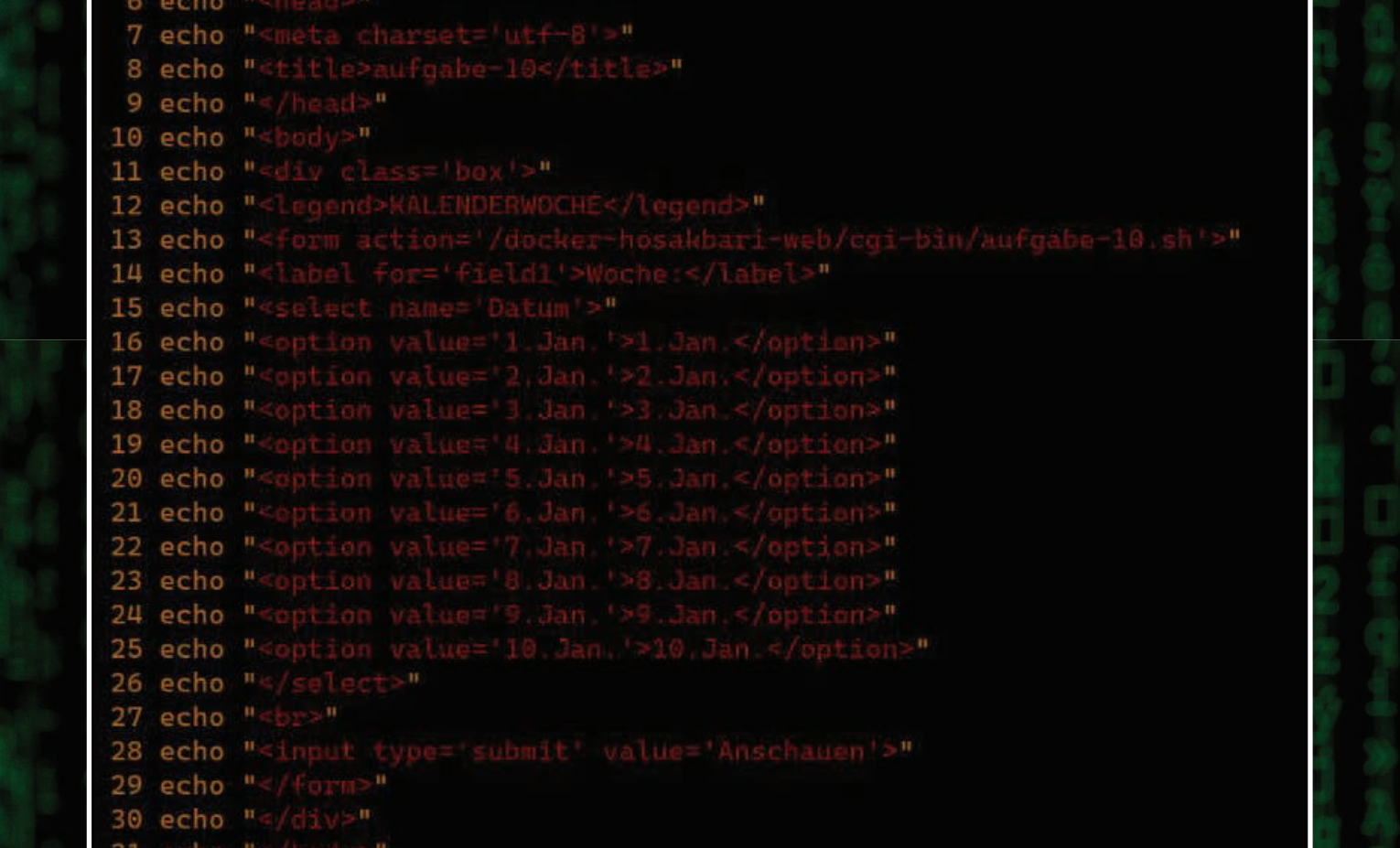
Kopiert Euch die Datei /home/step2022/data/rhodes.txt von hopper in Euren Docker-Container. Erstelle in Eurem Docker Container ein cgi-script, das unter https://informatikhs-bremerhaven.de/docker-DEINACCOUNT-web/cgi-bin/loesung-09.sh erreichbar ist. Das cgi-Script soll aus einer Anfrage der Form .../cgi-bin/loesung-09.sh?mmsi=<angefragtemmsi> eine gültige HTML-Antwort liefern, - in der alle Typ-1 Nachrichten - mit den Feldern 1,2,3,9 und 10 - für die angefragte MMSI - sortiert für die gesamte Zeile - nummeriert beginnend mit 1 - als HTML-Tabelle ohne Kopfzeile aufgeführt sind. Der HTML-Titel darf nur aus der angefragten MMSI bestehen. Wenn das Format der angefragten MMSI nicht genau einer Abfolge von exakt 9 Ziffern entspricht, darf der Inhalt des Body der HTML-Seite nur aus dem Text 'incorrect request' im Body bestehen.

```
1 |/bin/bash
2 echo "Content-type: text/html"
3 echo ""
4 echo "Content-type: text/html"
5 echo "Content-type: text/html"
6 echo "Content-type: text/html"
7 echo "Content-type: text/html"
8 echo "Content-type: text/html"
9 echo "Content-type: text/html"
10 echo "Content-type: text/html"
11 echo "Content-type: text/html"
12 echo "Content-type: text/html"
13 echo "Content-type: text/html"
14 echo "Content-type: text/html"
15 echo "Content-type: text/html"
16 echo "Content-type: text/html"
17 echo "Content-type: text/html"
18 echo "Content-type: text/html"
19 echo "Content-type: text/html"
20 echo "Content-type: text/html"
21 echo "Content-type: text/html"
22 echo "Content-type: text/html"
23 echo "Content-type: text/html"
24 echo "Content-type: text/html"
25 echo "Content-type: text/html"
26 echo "Content-type: text/html"
27 echo "Content-type: text/html"
28 echo "Content-type: text/html"
29 echo "Content-type: text/html"
30 echo "Content-type: text/html"
31 echo "Content-type: text/html"
32 echo "Content-type: text/html"
33 echo "Content-type: text/html"
```

## Aufgabe 10

erzeugt aus der Datei /home/step2022/rhodes-2022-12-17.txt für den 18.12.2022 eine Datei der Form: 1 58 00:00 2 57 00:01 3 53 00:02 wobei die zweite Spalte für die Häufigkeit von Typ-1-Nachrichten innerhalb der Minute des Tages in der dritten Spalte steht und die erste Spalte für die Nummer der Minute im Verlaufe des Tages. Diese Datei könnt Ihr mit einer einfachen Kombination aus grep, cut, uniq, -c, nl etc. erstellen, womit das Ergebnis sehr schnell erzeugt werden kann. Mit dieser Datei erzeugt dann eine Visualisierung mit gnuplot. Schreib eine HTML-Datei, in der in einem Formular ein Datum ausgesucht werden kann und erzeugt die Grafik in einem cgi-Skript.

```
1 |/bin/bash
2 cat /home/step2022/rhodes-2022-12-17.txt | grep "[1-5]" | cut -d " " -f 1 | cut -d " " -f 1 | cut -d " " -f 1 | cut -d " " -f 1
3 | uniq -c | awk -F " " '{print $2 "-" $3 "-" $4}' | sed "s/ /|/g" > loesung-10.txt
```



```
1 |/bin/bash
2 echo "Content-type: text/html"
3 echo ""
4 echo "Content-type: text/html"
5 echo "Content-type: text/html"
6 echo "Content-type: text/html"
7 echo "Content-type: text/html"
8 echo "Content-type: text/html"
9 echo "Content-type: text/html"
10 echo "Content-type: text/html"
11 echo "Content-type: text/html"
12 echo "Content-type: text/html"
13 echo "Content-type: text/html"
14 echo "Content-type: text/html"
15 echo "Content-type: text/html"
16 echo "Content-type: text/html"
17 echo "Content-type: text/html"
18 echo "Content-type: text/html"
19 echo "Content-type: text/html"
20 echo "Content-type: text/html"
21 echo "Content-type: text/html"
22 echo "Content-type: text/html"
23 echo "Content-type: text/html"
24 echo "Content-type: text/html"
25 echo "Content-type: text/html"
26 echo "Content-type: text/html"
27 echo "Content-type: text/html"
28 echo "Content-type: text/html"
29 echo "Content-type: text/html"
30 echo "Content-type: text/html"
31 echo "Content-type: text/html"
32 echo "Content-type: text/html"
33 echo "Content-type: text/html"
```

Man muss die Lösung in Gnuplot umwandeln