

Klausur

Computergrundlagen WS 2014/2015

JP Dr. Axel Arnold Tobias Richter Kai Kratzer
Julian Michalowsky Kai Szuttor Manuel Abele
Jonas Landsgesell Florian Fahrenberger

16. Februar 2015

Name	
Vorname	
Matrikelnummer	

Hinweise

- Die Maximalpunktzahl ist 100.
- Der verfügbare freie Platz gibt einen Hinweis darauf, welchen Umfang die Lösung haben sollte.
- Die Klausur ist sehr umfangreich, um alle Themengebiete abdecken zu können. In der Regel wird es nicht möglich sein, alle Aufgaben vollständig zu bearbeiten. Bearbeite deswegen zuerst die Themengebiete, die Dir besonders liegen!
- Falls der Platz nicht ausreichen sollte, verwende zusätzliche Blätter. Beschrifte diese unbedingt mit Deinem Namen und Deiner Matrikelnummer!
- Einige Fragen ähneln den Fragen aus vorigen Klausuren, sind aber *nicht* identisch! Lies die Fragen deshalb bitte *genau* durch!

Viel Erfolg!

1 Unixgrundlagen (20 Punkte)

Aufgabe 1:

(1 Punkt)

Was ist die wesentliche Funktion eines Betriebssystems?

Antwort:

|

Aufgabe 2:

(2 Punkte)

Was tut der folgende Shell-Befehl? Erkläre nicht die einzelnen Kommandos, sondern die Gesamtfunktion.

```
cd testdir && ls -l *.txt | wc -l
```

Antwort:

|

Aufgabe 3:

(2 Punkte)

Mit welchem Shell-Befehl lassen sich die Zeilen einer Datei `text.txt` zählen, die jeweils die Wörter `Professor` und `Axel` enthalten?

Antwort:

|

Aufgabe 4:

(3 Punkte)

Was tut der folgende Shell-Befehl? Erkläre nicht die einzelnen Teile, sondern die Gesamtfunktion.

```
ssh cgl14-001@cip0 "find_/var/log_-size_+50M"
```

Antwort:

|

Aufgabe 5:

(4 Punkte)

Markiere und benenne die vier Fehler im folgenden bash-Skript.

```
#!/bin/bash
filename= "backup- `date +%Y-%m-%d-%H-%M`.tar.gz"

if [ ! -d $HOME/backup ] then
    echo "Creating_backup_directory_in_$HOME/backup"
    mkdir -p $HOME/backup
fi

#Files and directories to exclude from backup
exclude="--exclude=$HOME/backup"

echo "Backing_up_$HOME_to_$HOME/backup/$filename"
tar $exclude -czf '$HOME/backup/$filename' $HOME

#check if archiving finished properly
if [ $? -eq 0 ]; then
    echo "Done"
else
    echo "Backup_may_be_broken"
```

Aufgabe 6:

(2 Punkte)

Was tut das obige Skript *nachdem die Fehler korrigiert wurden*?

Antwort:

|

Aufgabe 7:

(4 Punkte)

An einem Institutscomputer hat Benutzer kai folgenden Dialog in der Shell:

```
> groups kai tobias cgl14-001
kai : icp klausur sysguru stud video asm
tobias : icp klausur video
cgl14-001 : stud
> ls -la
total 16
drwxr-xr-x 4 kai icp      4096 Jan 23 11:39 .
drwxr-x--- 8 kai icp      4096 Jan 23 11:35 ..
drwxrwxr-x 2 kai icp      4096 Jan 23 11:36 cglstuff
-rw-r--r-- 1 kai asm          0 Jan 23 11:39 foo.dat
-rw-rw---- 1 kai klausur      0 Jan 23 11:39 klausur
-rw----- 1 kai icp          0 Jan 23 11:36 noten.txt
-rw-r--r-- 1 kai icp         43 Jan 23 14:25 skript.sh
```

Welcher der Benutzer kai, tobias, cgl14-001 kann welchen der folgenden Befehle erfolgreich ausführen?

```
cat klausur
rm -r cglstuff
```

Antwort:

|

Aufgabe 8:

(2 Punkte)

Der Benutzer kai versucht im oben aufgelisteten Verzeichnis Folgendes:

```
./skript.sh
```

Warum erhält er eine Fehlermeldung? Mit welchem Befehl kann er den Fehler beheben?

Antwort:

|

2 Python (25 Punkte)

Aufgabe 9: (2 Punkte)

Was ist der Hauptunterschied zwischen einer Liste (`list`) und einem Wörterbuch (`dict`) in Python? Nenne je einen Verwendungszweck!

Antwort:

|

Aufgabe 10: (3 Punkte)

Was ist der Hauptunterschied zwischen einer Interpretersprache (z.B. Python) und einer Compilersprache (z.B. C)?

Nenne jeweils einen Vor- und einen Nachteil!

Antwort:

|

Aufgabe 11:

(4 Punkte)

Das folgende Pythonprogramm hat verschiedene Fehler. Versuche zunächst zu verstehen, was das Programm wohl tun soll. Dann korrigiere die Fehler. Dabei handelt es sich um drei syntaktische Fehler und einen logischen Fehler.

```
int N = 100
zahlensieb = [0,0]

for i in range(2,N):
    zahlensieb.append(1)

# now remove true multiples
for i in range(2,N/2):
    if not zahlensieb[i]:
        continue # multiples already deleted
    multiple = 2*i
    while multiple < N:
        zahlensieb[multiple] = 0
        multiple += i

# print detected numbers
for i in range(2,N):
    if zahlensieb[i]:
        print i
```

Aufgabe 12:

(2 Punkte)

Was tut das obige Skript?

Antwort:

|

Aufgabe 13:

(2 Punkte)

Was ist die Ausgabe von folgendem Python-Dialog? Es geht nicht um die genaue Form, sondern darum, welche Elemente die Listen jeweils enthalten.

```
>>> Gruppe1 = ["Christian", "Axel", "Jens", "Maria"]
>>> Gruppe2 = Gruppe1[:]
>>> Gruppe3 = Gruppe1
>>> del Gruppe2[-1]
>>> del Gruppe3[1]
>>> print Gruppe1, Gruppe2
```

Antwort:

|

Aufgabe 14:

(8 Punkte)

Schreibe zunächst eine Python-Funktion `fak(k)`, die die Fakultät von k berechnet und zurückgibt.

Schreibe nun eine Python-Funktion `taylor_exp(x, n)`, die die Taylorentwicklung der Exponentialfunktion bis zur n -ten Ordnung mit Hilfe der Näherungsformel

$$\exp(x) \approx \sum_{i=0}^{i=n} \frac{x^i}{i!} \quad (1)$$

berechnet. Verwende hierzu die Funktion `fak(k)`.

Hinweis: Wenn Du die Funktion `fak(k)` nicht selber schreiben kannst, kannst Du trotzdem `taylor_exp(x, n)` schreiben!

Antwort:

|

Aufgabe 15:

(4 Punkte)

Ein Python-Skript berechnet die Wurzel nach dem Heron-Verfahren und bricht nach 4 Schritten ab. Das Skript wird nun durch folgende Zeilen erweitert:

```
import matplotlib.pyplot as plt

plt.subplot(2, 1, 1)
plt.plot(schritte, ergebnis, 'o-')
plt.ylabel('Berechneter_Wert')

plt.subplot(2, 1, 2)
plt.semilogy(schritte, abweichung, 'o-')
plt.xlabel('Iterationen')
plt.ylabel('Abweichung')

plt.show()
```

Die Listen mit den berechneten Ergebnissen enthalten dabei

```
schritte    = [0, 1, 2, 3, 4]
ergebnis    = [4.0, 2.5, 2.05, 2.0006097561, 2.0000000929]
abweichung  = [2.0e+0, 5.0e-1, 5.0e-2, 6.1e-4, 9.3e-08]
```

Skizziere die Abbildung, die mit `plt.show()` angezeigt wird!

Antwort:

|

3 C (20 Punkte)

Aufgabe 16:

(2 Punkte)

Wozu dient die folgende Kommandozeile?

```
gcc -o program program.c
```

Welche Option muss man noch hinzufügen, wenn im Programmcode Variablen direkt in Schleifenköpfen definiert werden sollen?

Antwort:

|

Aufgabe 17:

(1 Punkt)

Welches der beiden Programme output1 und output2 wird wahrscheinlich schneller laufen, wenn vorher folgende Zeilen aufgerufen werden? Warum?

```
gcc -O3 -o output1 program.c
gcc -O0 -o output2 program.c
```

Antwort:

|

Aufgabe 18:

(1 Punkt)

Mit der Zeile `#include <math.h>` wird in C eine Bibliothek für einfache mathematische Berechnungen eingebunden. Welche Zeile musst Du an den Anfang eines Programmes setzen, um später im Code Bildschirmausgaben mittels `printf()` machen zu können?

Antwort:

|

Aufgabe 19:

(4 Punkte)

Korrigiere die drei syntaktischen und den einen logischen Fehler in folgendem C-Programm:

```
//compile with gcc -std=c99 prog.c -lm -o prog -Wall
#include<math.h>
#include<stdio.h>
int main() {

    double change =0.00000001
    double estimate=2.1;
    double old_estimate=2.5;

    while(fabs(old_estimate-estimate)>change) {
        old_estimate=estimate;
        estimate=pow(1+1.0/n,n);
        print "iteration_%d,_estimate_%f\n" %(n, estimate);

        n+=1;
    }
    return 0;
}
```

Hinweis: fabs entspricht **abs** für den Datentyp float.

Aufgabe 20:

(1 Punkt)

Gib die Formel für die Folgenglieder an, die das vorhergehende Programm berechnet.

Antwort:

|

Aufgabe 21:

(4 Punkte)

Schreibe eine Funktion `area` in der Sprache C, die aus den drei Seitenlängen (a, b, c) eines Dreiecks dessen Fläche F_{Δ} nach dem Satz des Heron berechnet:

$$s = \frac{a + b + c}{2}$$
$$F_{\Delta} = \sqrt{s(s - a)(s - b)(s - c)}$$

Dabei kannst Du davon ausgehen, dass die Mathe-Bibliothek `math.h` bereits eingebunden ist. Die Funktion soll die drei Seitenlängen als Eingangsparameter vom Typ `double` erwarten und das Ergebnis auch als `double` zurückgeben.

Antwort:

|

Aufgabe 22:

(3 Punkte)

Welche Ausgabe erzeugt das folgende Programm?

```
#include<stdio.h>

int main(){
    char J[10]={'a','b','c','d','e','f','g','h','i','j'};
    printf("%c, %c, %c\n", J[0], J[1], J[2]);
    char* H=&J[1];
    printf("%c\n", *H);
    *H='5';
    printf("%c, %c, %c\n", J[0], J[1], J[2]);
    return 0;
}
```

Antwort:

|

Aufgabe 23:

(1 Punkt)

Schreibe eine Zeile C-Code, in der ein float array J dynamisch alloziert wird. Das Array soll die Größe 196 haben (d.h. 196 float Werte aufnehmen können).

Antwort:

|

Aufgabe 24:

(3 Punkte)

Schreibe einen Verbund (Struct) mit Typnamen „sample“. Dieser Struct soll zwei Zahlen vom Typ **double** mit den Namen `average` und `variance` aufnehmen. Erstelle anschließend eine Instanz dieses Structs und weise den einzelnen Feldern folgende Werte zu: `average=1.5` und `variance=0.5`.

Antwort:

|

4 Algorithmen und Datenstrukturen (10 Punkte)

Aufgabe 25:

(5 Punkte)

Beschreibe den Bubblesort-Algorithmus und diskutiere dessen Effizienz.

Antwort:

|

Aufgabe 26:

(1 Punkt)

Mit welcher Datenstruktur könnte man die Verzeichnisstruktur von unixartigen Betriebssystemen darstellen?

Antwort:

|

Aufgabe 27:

(4 Punkte)

Schreibe eine Funktion (in C)

```
int count_elements(struct list* liste)
```

die die Anzahl an Elementen in einer einfach verlinkten Liste zählt. Die Datenstruktur eines Knotens der verlinkten Liste sei wie folgt definiert:

```
struct list{
    int value;
    struct list* next;
};
```

Antwort:

|

5 L^AT_EX und Html (20 Punkte)

Aufgabe 28:

(2 Punkte)

Untenstehend findest du eine Formel in L^AT_EX-Quelltext. Schreibe diese Formel so, wie sie im kompilierten Dokument aussehen würde.

```
\begin{align}
  \sum_{k=0}^{\infty} q^k = \frac{1}{1-q}
  \quad \text{für } |q| < 1
\end{align}
```

Antwort:

|

Aufgabe 29:

(5 Punkte)

Korrigiere die vier Fehler in folgendem L^AT_EX-Dokument. Ergänze den Quelltext so, dass das Wort „Weltformel“ rot und fett ausgegeben wird.

```
\documentclass[a4paper]{scrartcl}

\begin{document}

In Abschnitt \ref{sec:weltformel} wird die
                Weltformel
(Gleichung \eqref{weltformel}) beschrieben.

\section{Die Superformel in \LaTeX \label{sec:weltformel}}

Im Jahre 1902 fand Kilian Effenberg aus Detmold folgende
Formel:

\begin{equation}
  W \cdot e^{\frac{1}{t}} = \sqrt{r} \cdot m_{\mathrm{el}}
  \label{eq:weltformel}
\end{align}

\end{document}
```

Aufgabe 30:

(6 Punkte)

Die folgende Tabelle wurde mit \LaTeX gesetzt:

Tabelle 1: Zahlensysteme: Verschiedene Repräsentationen derselben Zahl.

Zahl	Zahlensystem
7DF	hexadezimal
3737	oktal
11111011111	binär
2015	dezimal

Schreibe einen \LaTeX -Code, der diese Tabelle einschließlich der Überschrift erzeugen würde. Benutze logisches Markup!

Antwort:

|

Aufgabe 31:

(2 Punkte)

Nenne zwei grundlegende Unterschiede zwischen HTML und \LaTeX .

Antwort:

|

Aufgabe 32:

(5 Punkte)

Gegeben ist der folgende HTML-Quellcode:

```
<html>
<head>
<!-- <title>Eine wirklich schwierige Aufgabe</title> -->
  <style type="text/css">
    p {text-align:      right;}
    ul {font-weight:   bold;}
    h1 {font-style:    italic;}
  </style>
</head>
<body>
  <h1>Aufgabe</h1>
  <p>Ziel ist das Verst&auml;ndnis dieses Quelltextes.</p>
  <ol>
    <li>item 1</li>
    <li>item 2</li>
  </ol>
  <p>Tabellen sind der Wahnsinn:</p>
  <table align="center" border="1">
    <tr> <td>Tabellen</td> <td>sind</td> </tr>
    <tr> <td>der</td> <td>Wahnsinn</td> </tr>
  </table>
</body>
</html>
```

Beschreibe/skizziere die Ausgabe, die der Browser aus diesem Quelltext erzeugt.

Antwort:

|

6 Visualisierung (5 Punkte)

Aufgabe 33:

(2 Punkte)

Ergänze folgendes Gnuplot-Skript, um eine Gerade an die gegebenen Datenpunkte zu fitten. Das resultierende Skript soll Datenpunkte und Fitfunktion in einem Plot darstellen.

Hinweis: Wenn Du nicht weißt, wie man in gnuplot fittet, kannst Du trotzdem die Funktionen plotten. Nenne das Ergebnis des Fits dann einfach `fit(x)`.

```
>plot "Data.txt" u 1:3 w lp title "Messwerte"
```

```
>
```

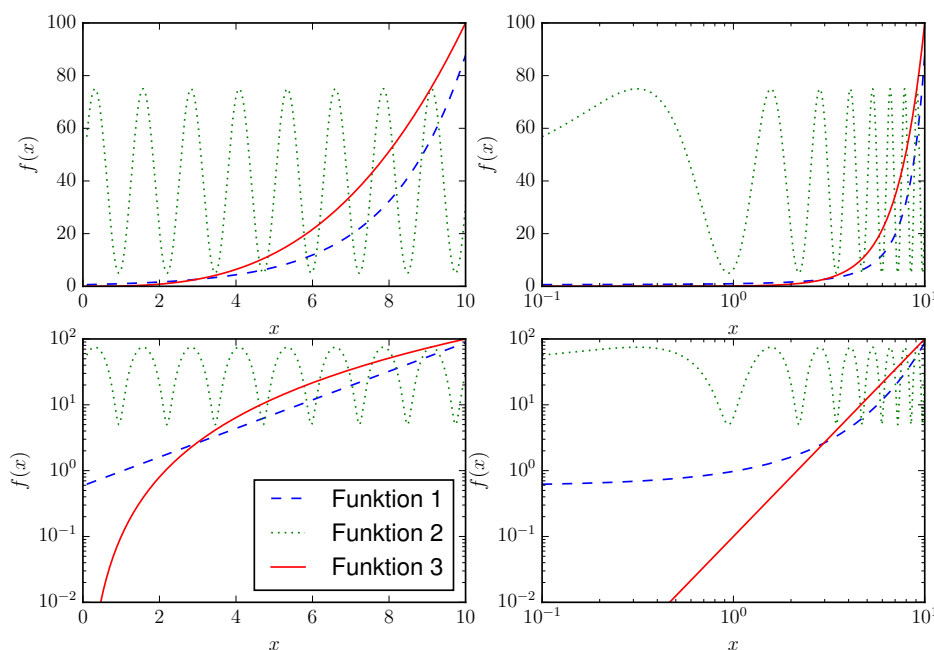
```
>
```

```
>
```

Aufgabe 34:

(3 Punkte)

Folgende Plots stellen drei Funktionen auf verschiedene Arten dar. Gib zu jeder Kurve eine Funktion an, die diese bestmöglich beschreibt. Parameter sind nicht verlangt! Beispiel: Beschreibe eine vermeintlich logarithmische Funktion $f_1(x)$ nur mit $f_1(x) \propto \log(x)$.



Antwort:

|