

Probeklausur

Computergrundlagen WS 2012/2013

JP Dr. Maria Fyta JP Dr. Axel Arnold Elena Minina
Florian Weik Kai Kratzer Rudolf Weeber
Stefan Kesselheim Tobias Richter

8. Februar 2013

Name	
Vorname	
Matrikelnummer	

Hinweise

- Der verfügbare freie Platz gibt einen Hinweis darauf, welchen Umfang die Lösung haben sollte.
- Dies ist eine Probeklausur, daher sind einige Aufgaben identisch mit Aufgaben aus den Klausuren vorheriger Jahrgänge. Dies wird in der tatsächlichen Klausur *nicht* der Fall sein! Allerdings können einige *Aufgabenstellungen* sehr ähnlich aussehen, daher in der Klausur bitte auf den genauen Text achten!
- Die Maximalpunktzahl ist 60, genau wie das auch in der echten Klausur der Fall sein wird. Auch die Verteilung auf die Themenbereiche ist vergleichbar.

Viel Erfolg!

1 Unixgrundlagen (10 Punkte)

Aufgabe 1: (1 Punkt)

Zähle 3 Dienste im Internet auf.

Antwort:

| WWW, Email, Usenet, FTP, Telnet, SSH, ...

Aufgabe 2: (1 Punkt)

Mit welchem Shell-Befehl kann man sämtliche Dateien aus dem aktuellen Verzeichnis in das Unterverzeichnis *newdir* *verschieben*, die die Endung `.txt` haben?

Antwort:

| `mv *.txt newdir/`

Aufgabe 3: (1 Punkt)

Wie ruft man die Hilfeseite des Shell-Befehls `rm` auf?

Antwort:

| `man rm`

Aufgabe 4: (1 Punkt)

Was tut der Befehl `„medit notes.txt &“` ?

Antwort:

| Der Befehl ruft das Programm `medit` im Hintergrund auf, übergibt `„notes.txt“` als Argument und kehrt danach zur Shell zurück.

Aufgabe 5: (1 Punkt)

Was tut der Befehl `mv oldnotes* /tmp?`

Antwort:

|

Aufgabe 6: (1 Punkt)

Was tut der Befehl `find . -size -300M?`

Antwort:

|

Aufgabe 7:

(1 Punkt)

Was tut der Befehl `./backup.sh -now`?

Antwort:

|

Aufgabe 8:

(1 Punkt)

Was tut der Befehl `scp notes.txt horst@icp.uni-stuttgart.de:Notes.txt`?

Antwort:

|

Aufgabe 9:

(1 Punkt)

Was tut der Befehl `kill 42`?

Antwort:

|

Aufgabe 10:

(1 Punkt)

Mit welchem Befehl kann man sich alle Zeilen der Datei `gpl.txt` ausgeben lassen, die sowohl das Wort "freedom" als auch das Wort "software" enthalten?

Antwort:

|

2 Permissions (5 Punkte)

Auf einem Unix-Rechner gibt Benutzer `axel` die Befehle `groups axel chris dirk bob` und `ls -la` in einer Shell ein und erhält folgende Ausgabe:

```
$ groups axel chris dirk
axel   : icp cgl1112 dozent
chris  : icp cgl1112
dirk   : cgl1112
bob    : user
$ ls -la
total 4
drwxrwxr-x  2 axel icp          60 2010-10-27 13:23 .
drwxr-xr-x 22 axel icp       4096 2010-10-27 13:22 ..
-rwxr----- 1 chris  cgl1112    0 2010-10-27 13:23 file.sh
-rw-r----- 1 axel  icp          0 2010-10-27 13:22 musterloesungen.txt
```

Aufgabe 11: (1 Punkt)

Welcher der Benutzer axel, chris, dirk und bob kann die Datei `file.sh` lesen?

Antwort:

| axel, chris, dirk

Aufgabe 12: (1 Punkt)

Welcher der Benutzer axel, chris, dirk und bob kann die Datei `file.sh` löschen?

Antwort:

| axel, chris

Aufgabe 13: (1 Punkt)

Welcher der Benutzer axel, chris, dirk und bob kann die Datei `file.sh` ausführen?

Antwort:

| chris

Aufgabe 14: (1 Punkt)

Wie sehen die Rechte der Datei `file.sh` aus, nachdem `chris` den Befehl `chmod a+x file.sh` ausgeführt hat?

Antwort:

| -rwxr-x--x

Aufgabe 15: (1 Punkt)

Was muss `axel` machen, damit nur noch er die beiden Dateien ansehen kann?

Antwort:

| `chmod og-r *`

3 Python (13 Punkte)

Aufgabe 16:

(2 Punkte)

Was gibt der folgende Python-Befehl aus, und warum?

```
print 4/3
```

Antwort:

Der Befehl gibt 1 aus. Zähler und Nenner sind ohne Dezimalpunkt, also als Integers definiert. Dementsprechend ist das Ergebnis für Python auch ein Integer. Es wird bei der Rechnung nicht gerundet, sondern abgeschnitten, also wird aus 1.33.. eine Eins.

Aufgabe 17:

(1 Punkt)

Wie erhält Du ein naheliegenderes Ergebnis?

Antwort:

```
print 4./3
```

Aufgabe 18:

(3 Punkte)

Betrachte das folgende Pythonprogramm:

```
fac = 1
while i in range(1,21):
    fac == fac*i

print "fac(20)␣=", Fak
```

Das Programm enthält ein paar Fehler. Schreibe hier das korrigierte Programm hin.

Antwort:

```
fac = 1
for i in range(1,21):
    fac = fac*i

print "fac(20)␣=", fac
```

Aufgabe 19:

(1 Punkt)

Was tut das korrigierte Programm der vorigen Aufgabe?

Antwort:

|

Aufgabe 20:

(1 Punkt)

Betrachte die folgende Pythonfunktion:

```
def p(e, n):  
    if n == 0: return 1  
    else return e*p(e, n-1)
```

Welche mathematische Funktion berechnet die Pythonfunktion?

Antwort:

|

Aufgabe 21:

(3 Punkte)

Die Sequenz der Fibonacci-Zahlen ist wie folgt definiert:

$$\text{fib}(n) = \begin{cases} 1 & \text{falls } n \leq 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{sonst} \end{cases} \quad (1)$$

Schreibe eine Pythonfunktion `fib`, die die n -te Zahl der Fibonacci-Sequenz wie in der oben gegebenen Definition berechnet.

Hinweis Eine Pythonfunktion wird wie folgt definiert:

```
def fib(n):  
    ...
```

Antwort:

```
def fib(n):  
    if n <= 1: return 1  
    else: return fib(n-1) + fib(n-2)
```

|

Aufgabe 22: (1 Punkt)

Ist die Funktion `fib` aus der vorigen Aufgabe *rekursiv* oder *iterativ* definiert?

Antwort:

|

Aufgabe 23: (1 Punkt)

Wie oft wird die Funktion `fib` aufgerufen, wenn man `fib(3)` ausführt?

Antwort:

| 5 mal.

4 Boole'sche Algebra (2 Punkte)

Aufgabe 24: (1 Punkt)

Gegeben sei der Boole'schen Ausdruck $F = \neg a \vee \neg(b \vee a)$.

Stelle für F eine Wertetafel mit jeweils allen Belegungen der Variablen a und b auf.

Antwort:

	$a = 0$	$a = 1$
$b = 0$	1	0
$b = 1$	1	0

Aufgabe 25: (1 Punkt)

Lies aus der Wertetabelle in der vorigen Aufgabe einen vereinfachten Ausdruck für F ab.
Wie lautet er?

Antwort:

| $F = \neg a$

5 Zahlensysteme (5 Punkte)

Aufgabe 26: (3 Punkte)

Rechne die Dezimalzahl 42 von Hand (u.U. unter Zuhilfenahme der nebenstehenden Tabelle) in das Hexadezimal-, das Binär- und das Septalsystem (zur Basis 7) um. Notiere den Lösungsweg!

Antwort:

|

	2	8	10	16
0	0	0	0	0
1	1	1	1	1
10	2	2	2	2
11	3	3	3	3
100	4	4	4	4
101	5	5	5	5
110	6	6	6	6
111	7	7	7	7
1000	10	8	8	8
1001	11	9	9	9
1010	12	10	A	A
1011	13	11	B	B
1100	14	12	C	C
1101	15	13	D	D
1110	16	14	E	E
1111	17	15	F	F
10000	20	16	10	10

Aufgabe 27: (1 Punkt)

Rechne die Hexadezimalzahl BAE von Hand (u.U. unter Zuhilfenahme der obenstehende Tabelle) in das Binär- und Oktalsystem um.

Antwort:

|

Aufgabe 28: (1 Punkt)

In welchem Zahlensystem rechnet ein Computer? Warum?

Antwort:

| Im binären System. Eine binäre Stelle kann elektrisch einfach durch die Zustände "Spannung an" und "Spannung aus" dargestellt werden.

6 Fließkommazahlen und numerische Fehler (5 Punkte)

Aufgabe 29: (1 Punkt)

Wie ist eine Fließkommazahl aufgebaut?

Antwort:

| Eine Fließkommazahl besteht aus Vorzeichen, Mantisse m , und Exponent e . Der numerische Wert ist von der Form $0, m \times b^e$, wobei b die zugrundeliegende Basis ist.

Aufgabe 30:

(2 Punkte)

Betrachte das folgende Python-Programm:

```
e = 1
while True:
    a = (1.0 + 1.0/10.0**e) - 1.0
    if a == 0.0: break
    e += 1
print e
```

Was ist der vom Programm ausgegebene Wert e ? Wie groß ist dieser auf einem gewöhnlichen PC etwa?

Antwort:

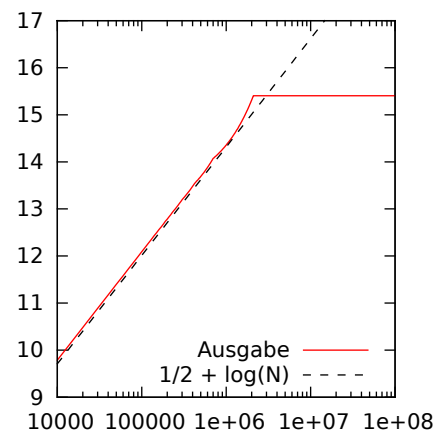
Das Programm gibt die Anzahl der Stellen der Mantisse an, also 16 auf einem normalen PC.

Aufgabe 31:

(2 Punkte)

Die Reihe $\sum_{i=k}^N \frac{1}{k}$ verhält sich etwa wie $\frac{1}{2} + \log(N)$, divergiert also. Folgendes C-Programm nähert diese Summe numerisch:

```
#include <stdio.h>
int main() {
    float s = 0;
    for (int j = 1; j < 1e8; j++) {
        s += 1.0/j;
        if (j % 10000 == 0)
            printf("%d_%f\n", j, s);
    }
    return 0;
}
```



Stimmt die Mathematik nicht? Oder ist das Programm vielleicht falsch? Warum „konvergiert“ die numerische Summe? Was würde passieren, wenn statt `float` der Datentyp `double` verwendet würde?

Antwort:

Das Programm ist in Ordnung, aber kann nur zum Summieren von Termen bis zu etwa 10^{-6} genutzt werden, da dann die Auflösung des `float`-Datentyps erreicht ist. Mit `double` ginge es bis 10^{-16} .

7 L^AT_EX (8 Punkte)

Aufgabe 32:

(2 Punkte)

Beschreibe die Unterschiede zwischen *Layout*, *Struktur* und *Inhalt* eines Textes anhand des folgenden Textstückes:

Verbot

Es ist *nicht* erlaubt, von anderen Klausurteilnehmern abzuschreiben. Zumindest aber ist es verboten, sich dabei erwischen zu lassen.

Antwort:

|

Aufgabe 33:

(2 Punkte)

Welcher der folgenden L^AT_EX-Befehle ist visuelles, welcher logisches Markup?

- `\emph{...}` Antwort: logisch
- `\textit{...}` Antwort: visuell
- `\textbf{...}` Antwort: visuell
- `\begin{enumerate} ... \end{enumerate}` Antwort: logisch
- `\begin{figure} ... \end{figure}` Antwort: logisch
- `\subsection{...}` Antwort: logisch

Aufgabe 34:

(3 Punkte)

Der Text im folgenden Kasten wurde mit Hilfe von L^AT_EX gesetzt:

1. Klausurtermine
 - Probeklausur: 1.4.2012
 - *echte* Klausur: **2.4.2012**
2. Einsicht: 1.4.2013
3. Bestehen bei $\frac{1}{2}$ der Gesamtpunktzahl

Ergänze den folgenden Text durch L^AT_EX-Befehle so, daß sich die in der obigen Box gezeigte Ausgabe ergeben würde.

```
\begin{enumerate}
\item Klausurtermine
  \begin{itemize}
    \item Probeklausur: 1.4.2012
    \item \emph{echte} Klausur: \textbf{2.4.2012}
  \end{itemize}
\item Einsicht: 1.4.2013
\item Bestehen bei $\frac{1}{2}$
      der Gesamtpunktzahl
\end{enumerate}
```

Aufgabe 35:

(1 Punkt)

Wie kannst Du die folgende Formel in L^AT_EX setzen?

$$e^{\pi i} - 1 = 0 \quad (2)$$

Antwort:

```
\begin{equation}
e^{\mathbf{\pi} i} - 1 = 0
\end{equation}
```

8 Bildbearbeitung (2 Punkte)

Aufgabe 36: (1 Punkt)

Du speicherst das Foto von Axel im JPG- und im PNG-Format, jeweils mit mittlerem Kompressionsgrad. Welche der Dateien ist vermutlich kleiner?

Antwort:

|

Aufgabe 37: (1 Punkt)

Du speicherst einen Funktionsplot im SVG- und im JPG-Format. Welche der Dateien eignet sich besser zum Ausdruck im Großformat, und warum?

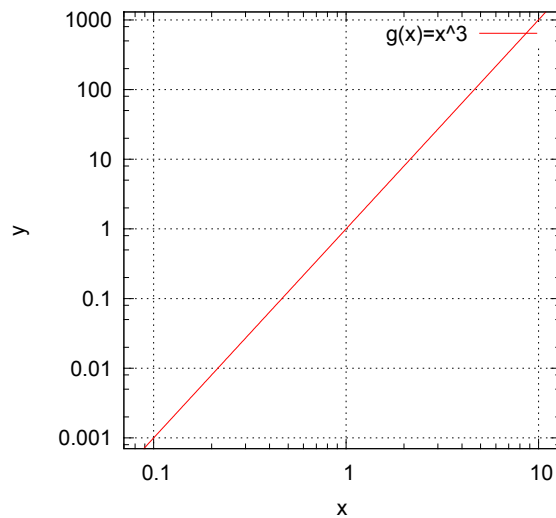
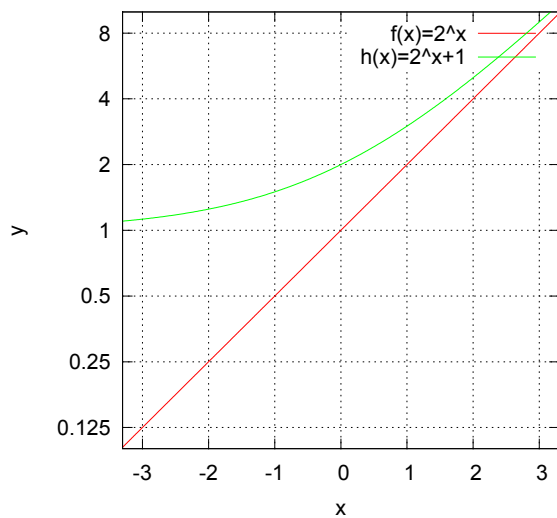
Antwort:

|

9 Gnuplot (4 Punkte)

Aufgabe 38: (2 Punkte)

Skizziere die Funktionen $f(x) = 2^x$, $g(x) = x^3$ und $h(x) = 2^x + 1$ im jeweils am besten dafür geeigneten der untenstehenden Graphen. Vergiss nicht, die Funktionen zu beschriften!



Aufgabe 39: (1 Punkt)

Welches Kommando musst Du in gnuplot benutzen, um die Funktionen zu zeichnen? In der Legende sollen sie mit „f(x)“, „g(x)“ und „h(x)“ erscheinen.

Antwort:

```
plot 2**x title "f(x)", x**3 title "g(x)", \
     2**x+1 title "h(x)"
```

Aufgabe 40:

(1 Punkt)

Welche Gnuplot-Kommandos benötigst Du, um die im linken Graphen benutzten Achsenbereiche, -skalierungen und -beschriftungen zu erzeugen?

Antwort:

```
set xrange [-3.3:3.3]
set yrange [0.1:8.5]
set logscale y
```

10 C (6 Punkte)

Aufgabe 41:

(1 Punkt)

Warum ist der Betriebssystemkern von Linux nicht in der Programmiersprache Python, sondern in den Programmiersprachen C und Assembler implementiert?

Antwort:

|

Aufgabe 42:

(2 Punkte)

Schreibe die folgende `for`-Schleife in eine `while`-Schleife um, die dasselbe tut

```
for (int i=0; i < 10; i++) {  
    if (exp(i) > 400) break;  
}
```

Antwort:

```
int i = 0;  
while (i < 10) {  
    if (exp(i) > 400) break;  
    i++;  
}
```

Aufgabe 43:

(3 Punkte)

Welche Ausgabe erzeugt das folgende C-Programm?

```
#include <stdio.h>  
int main(){  
    char x[] = "Hallo_Welt!\n";  
    printf("%s", x);  
    char *y = &(x[5]);  
    *y = 0;  
    x[1] = 'e';  
    printf("%s_Olaf!\n", x);  
}
```

Antwort:

```
|Hallo Welt!  
|Hello Olaf!
```