

Probeklausur

Computergrundlagen WS 2010/2011

JP Dr. Axel Arnold Dr. Olaf Lenz Florian Rühle
Thomas Zauner Shervin Rafatnia Kai Kratzer
Rudolf Weeber

8. Februar 2011

Name	
Vorname	
Matrikelnummer	

Hinweise

- In der Regel gibt der verfügbare freie Platz einen Hinweis darauf, welchen Umfang die Lösung haben sollte.
- Falls der Platz nicht ausreichen sollte, verwende zusätzliche Blätter. Beschrifte diese mit Deinem Namen und Matrikelnummer!
- Die Maximalpunktzahl ist 60.

1 Unixgrundlagen (10 Punkte)

Aufgabe 1:

(1 Punkt)

Was ist der Unterschied zwischen einem Prozess und einem Programm?

Antwort:

Aufgabe 2:

(1 Punkt)

Zähle 3 Dienste im Internet auf.

Antwort:

Aufgabe 3:

(1 Punkt)

Worin unterscheiden sich die folgenden beiden Befehle?

```
cat gpl.txt > new.txt
cat gpl.txt >> new.txt
```

Antwort:

Aufgabe 4:

(1 Punkt)

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

Antwort:

Aufgabe 5:

(1 Punkt)

Was tut der Befehl `man cp`?

Antwort:

Aufgabe 6:

(1 Punkt)

Was tut der Befehl `grep -i computergrundlagen notes.txt`?

Antwort:

Aufgabe 7:

(1 Punkt)

Was tut der Befehl `emacs notes.txt &`?

Antwort:

Aufgabe 8:

(1 Punkt)

Was tut der Befehl `./myscript`?

Antwort:

Aufgabe 9:

(1 Punkt)

Was tut der Befehl `chown olenz:icp notes.txt`?

Antwort:

Aufgabe 10:

(1 Punkt)

Was tut der Befehl `ssh horst@icp.uni-stuttgart.de`?

Antwort:

2 Permissions (4 Punkte)

Auf einem Unix-Rechner gibt Benutzer `olenz` die Befehle `groups olenz floh cgl1355` und `ls -la` in einer Shell ein und erhält folgende Ausgabe:

```
$ groups olenz floh cgl1355
olenz   : icp cgl1011 dozent
floh    : icp cgl1011
cgl1355 : cgl1011
bob     : user
$ ls -la
total 4
drwxrwxr-x  2 olenz icp      60 2010-10-27 13:23 .
drwxr-xr-x 22 olenz icp    4096 2010-10-27 13:22 ..
-rw-r--r--  1 olenz cgl1011   0 2010-10-27 13:23 bla.txt
-rwxrw----  1 floh  dozent    0 2010-10-27 13:23 file.sh
-rw-r-----  1 olenz icp      0 2010-10-27 13:22 musterloesungen.txt
```

Aufgabe 11: (1 Punkt)

Welcher der Benutzer `olenz`, `floh`, `cgl1355` und `bob` kann die Datei `musterloesungen.txt` lesen?

Antwort:

Aufgabe 12: (1 Punkt)

Welcher der Benutzer `olenz`, `floh`, `cgl1355` und `bob` kann eine neue Datei in diesem Verzeichnis anlegen?

Antwort:

Aufgabe 13: (1 Punkt)

Welcher der Benutzer `olenz`, `floh`, `cgl1355` und `bob` kann die Datei `file.sh` ausführen?

Antwort:

Aufgabe 14: (1 Punkt)

Welchen Befehl muss `floh` ausführen, um dafür zu sorgen, daß `olenz` die Datei `file.sh` ausführen kann?

Antwort:

3 Turingmaschine (4 Punkte)

Eine Turingmaschine ($\Gamma = \{_, 1\}$, $Z = \{A, B, C, D\}$) benutzt die folgende Übergangstabelle:

state	read	write	move	next state
A	-	-	→	A
A	1	1	→	B
B	-	-	←	C
B	1	1	→	B
C	-	-	←	D
C	1	-	←	D
D	-	-	←	STOP
D	1	1	←	D

Das Eingabeband enthält dabei die Zeichenkette „..._111_...“ und der Lese-/Schreibkopf ist auf dem ersten „_“ ganz links positioniert. Der Anfangszustand ist A.

Aufgabe 15: (3 Punkte)

Simuliere die Turingmaschine! Schreibe dazu die folgende Tabelle fort. Rahme die Position des Lese-/Schreibkopfes ein.

Zustand	Band					
A	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="border: 2px solid black;">_</td><td>1</td><td>1</td><td>1</td><td>_</td></tr></table>	_	1	1	1	_
_	1	1	1	_		
A	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="border: 2px solid black;">_</td><td style="border: 2px solid black;">1</td><td>1</td><td>1</td><td>_</td></tr></table>	_	1	1	1	_
_	1	1	1	_		

Aufgabe 16: (1 Punkt)

Was ist das Ergebnis der Berechnung? Was tut das Programm? (Hinweis: Zahlen werden bei dieser Turingmaschine im unären Zahlensystem notiert, d.h. eine „3“ wird als „111“ notiert, eine „5“ als „11111“)

Antwort:

4 Python (11 Punkte)

Die Fakultät ist wie folgt rekursiv definiert:

$$n! = \begin{cases} 1 & \text{falls } n \leq 1 \\ n(n-1)! & \text{sonst} \end{cases}$$

Hinweis Eine Pythonfunktion wird wie folgt definiert:

```
def fac(n):  
    print(n)
```

Aufgabe 17:

(3 Punkte)

Schreibe eine Pythonfunktion `fac`, das die Fakultät wie in der Formel rekursiv berechnet.

Aufgabe 18:

(3 Punkte)

Schreibe eine Pythonfunktion, die die Fakultät iterativ (d.h. mit Hilfe einer Schleife) berechnet.

Aufgabe 19:

(2 Punkte)

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

```
print 3/4
```

Antwort:

Betrachte das folgende Pythonprogramm:

```
number==0
while Number < 100
    if number < 50
        print "n_ist_kleiner_als_50:", number
    else
        print "n_ist_groesser_gleich_50:", number
number+=1
```

Aufgabe 20:

(2 Punkte)

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

Aufgabe 21:

(1 Punkt)

Was tut das korrigierte Programm der vorigen Aufgabe?

Antwort:

5 Asymptotisches Verhalten (6 Punkte)

Aufgabe 22:

(2 Punkte)

Angenommen, eine Gruppe von n Leuten teilen sich einen Kuchen. Von welcher Ordnung ($\mathcal{O}_{n \rightarrow \infty}(\cdot)$) ist die Größe (Gewicht) des Kuchenstückes pro Person?

Antwort:

Aufgabe 23:

(2 Punkte)

Ordne die folgenden Funktionen einer der Ordnungen

$\mathcal{O}_{x \rightarrow \infty}(x)$, $\mathcal{O}_{x \rightarrow \infty}(x^2)$, $\mathcal{O}_{x \rightarrow \infty}(2^x)$, $\mathcal{O}_{x \rightarrow \infty}(\frac{1}{x})$ oder $\mathcal{O}_{x \rightarrow \infty}(x \log x)$ zu.

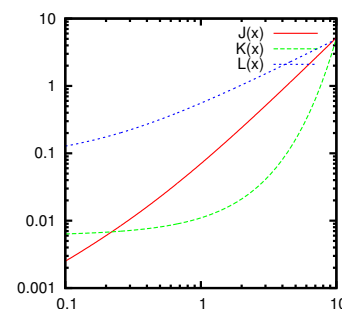
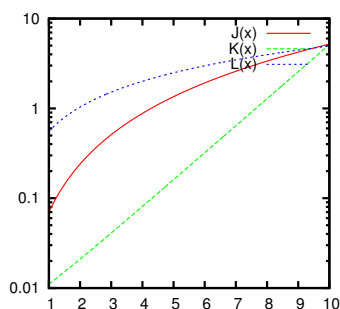
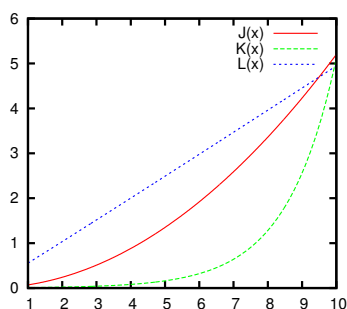
- $f(x) = \frac{1}{1000} + \frac{1}{200} \times 2^x$
- $g(x) = -0.9 + \frac{1}{\log(x+8)} + \frac{1}{2}(x+1)$
- $h(x) = \frac{1}{200}x^2 + \frac{1}{50}x$

Antwort:

Aufgabe 24:

(2 Punkte)

In den folgenden Graphen sind die Funktionen aus der vorigen Aufgabe geplottet. Welche der Funktionen entspricht welchem Plot?



6 Boole'sche Algebra (5 Punkte)

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

Aufgabe 25: (1 Punkt)
Stelle für F eine Wertetafel mit jeweils allen Belegungen der Variablen a und b auf.

Antwort:

Aufgabe 26: (1 Punkt)
Welche Aufgabe erfüllt in diesem Zusammenhang das folgende Pythonskript?

```
for a in [True, False]:
    for b in [True, False]:
        F = not (not (a or b) or (a and not b))
        print("a=%s b=%s F=%s" % (a, b, F))
```

Antwort:

Aufgabe 27: (3 Punkte)
Vereinfache den Ausdruck F so lange, bis keines der Gesetze der boole'schen Logik mehr anwendbar ist. Notiere bei jedem Rechenschritt, welches Gesetz verwendet wurde!

- $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ (1)
- $a \vee (b \vee c) = (a \vee b) \vee c$ (2)
- $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ (3)
- $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ (4)
- $a \wedge b = b \wedge a$ (5)
- $a \vee b = b \vee a$ (6)
- $a \wedge (a \vee b) = a$ (7)
- $a \vee (a \wedge b) = a$ (8)
- $a \wedge \neg a = 0$ (9)
- $a \vee \neg a = 1$ (10)
- $a \vee 0 = a$ (11)
- $a \wedge 1 = a$ (12)
- $a \vee a = a$ (13)
- $a \wedge a = a$ (14)
- $a \vee 1 = 1$ (15)
- $a \wedge 0 = 0$ (16)
- $\neg \neg a = a$ (17)
- $\neg 0 = 1$ (18)
- $\neg 1 = 0$ (19)
- $\neg(a \vee b) = \neg a \wedge \neg b$ (20)
- $\neg(a \wedge b) = \neg a \vee \neg b$ (21)

7 Zahlensysteme (4 Punkte)

Aufgabe 28: (3 Punkte)

Rechne die Hexadezimalzahl ABBA von Hand (u.U. unter Zuhilfenahme der nebenstehenden Tabelle) in das Binär-, Dezimal-, und Oktalsystem um. Notiere den Lösungsweg!

	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	10	A
1011	13	11	11	B
1100	14	12	12	C
1101	15	13	13	D
1110	16	14	14	E
1111	17	15	15	F
10000	20	16	16	10

Aufgabe 29:

(1 Punkt)

In welchem Zahlensystem rechnet ein Computer? Warum?

Antwort:

Aufgabe 30:

(1 Punkt)

Welchen Vorteil bietet das Hexadezimalsystem gegenüber dem Dezimalsystem im Computereinsatz?

Antwort:

Aufgabe 31:

(2 Punkte)

Wenn man in Python die Rechnung $600000000000.0 * (1.0 / 600000000000.0)$ ausführt, dann erhält man als Ergebnis 0.9999999999999999 . Wieso ist das Ergebnis nicht 1? Ist der Prozessor kaputt?

Antwort:

8 L^AT_EX(5 Punkte)

Aufgabe 32:

(1 Punkt)

Was ist der Unterschied zwischen visuellem und logischem Markup?

Antwort:

Aufgabe 33:

(2 Punkte)

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

- `\textit{...}`
- `\textbf{...}`
- `\emph{...}`
- `\begin{center} ... \end{center}`
- `\begin{itemize} ... \end{itemize}`
- `\section{...}`

Aufgabe 34:

(2 Punkte)

Die folgende Tabelle wurde mit L^AT_EX gesetzt.

Name	Telefon	Raum
Olaf Lenz	63607	209
Axel Arnold	67609	201

Ergänze das folgende Stück von L^AT_EX-Code so, daß es die Tabelle erzeugen würde.

```
\begin{tabular}{|l|r|r|}
\hline
\textbf{Name} & & \\
\hline
\end{tabular}
```

9 Bildbearbeitung (2 Punkte)

Aufgabe 35:

(2 Punkte)

Welches Grafikformat (PNG, JPG oder SVG) eignet sich am besten für die folgenden Grafiktypen, und warum?

- Fotos im WWW
- Mathematische Plots zur Verwendung in einem (wissenschaftlichen) Schriftstück
- einfache Diagramme zur Veröffentlichung auf einer Webseite

Antwort:

10 Gnuplot (2 Punkte)

Aufgabe 36:

(2 Punkte)

Die Datei `histo.dat` enthalte die Daten einer mittels Zufallszahlen erzeugten verrauschten Normalverteilung. Skizziere den vom folgenden Gnuplot-Skript erzeugten Plot.

```
g(x) = A*exp(-B*(x-m)**2)
fit g(x) "histo.dat" via A,B,m
set xrange [-5,5]
plot "histo.dat" with yerrorbars, g(x)
```

11 C (7 Punkte)

Aufgabe 37:

(1 Punkt)

Was tut der Interpreter einer Programmiersprache?

Antwort:

Aufgabe 38:

(3 Punkte)

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

```
int i=0;
while (i < 35)
{
    printf("%d\n", fib(i));
    i++;
}
```

Aufgabe 39:

(3 Punkte)

Welche Ausgabe erzeugt das folgende C-Programm?

```
#include <stdio.h>
int main()
{
    int a[3];
    int *b = &a[2];

    for (int i = 0; i < 3; i++)
        a[i] = i*i;
    printf("%d_%d_%d\n", a[0], a[1], a[2]);
    *b /= 2;

    printf("%d_%d_%d\n", a[0], a[1], a[2]);
}
```