

Übungsblatt 6: Python und Zahlensysteme

22. November 2011

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Dienstag, 29.11., 13:00** für die Übungsgruppen am Mittwoch und Donnerstag
- **Donnerstag, 1.12., 13:00** für die Übungsgruppen am Montag und Dienstag

Die Lösungen solltest Du in eine Kopie der Datei `/share/Courses/CG2011/06/vorlage06.txt` einfügen. Zur Abgabe schickst Du die Lösungsdatei im Anhang einer Email an Deinen Tutor.

Aufgabe 6.1: Zahlensysteme (3 Punkte)

- 6.1.1 (2 Punkte) Berechne die folgenden Zahlen a bis k , indem Du zwischen verschiedenen Zahlensystemen umrechnest. Dabei steht 1234_7 für die Zahl 1234 im Zahlensystem zur Basis 7. $a_{10} = 1234_7$ bedeutet also, daß die Zahl 1234 im Zahlensystem zur Basis 7 ins Zahlensystem zur Basis 10 (Dezimalsystem) umgerechnet werden soll.

Hinweis In einigen (aber nicht allen) Fällen dürfte die nebenstehende Tabelle nützlich sein.

- $a_{10} = 1234_7$
- $b_7 = 1234_{10}$
- $c_{16} = 4321_7$
- $d_{16} = 10000001_2$
- $e_{16} = 10100101_2$
- $f_8 = 10000001_2$
- $g_8 = 10100101_2$
- $h_2 = CD_{16}$
- $i_2 = 27_8$

2	7	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	10	7	7	7
1000	11	10	8	8
1001	12	11	9	9
1010	13	12	10	A
1011	14	13	11	B
1100	15	14	12	C
1101	16	15	13	D
1110	20	16	14	E
1111	21	17	15	F
10000	22	20	16	10

- 6.1.2 (1 Punkt) Im Computerumfeld wird häufig das Hexadezimalsystem ($B = 16$) verwendet. Welchen Vorteil bietet das System gegenüber dem Dezimalsystem ($B = 10$) im Computerumfeld? Welchen Vorteil bietet es gegenüber dem Oktalsystem ($B = 8$)?

Aufgabe 6.2: Python: Dictionaries, Strings und Module (7 Punkte)

- 6.2.1 (3 Punkte) Schreibe ein Skript `occurrence.py`, das folgenden Spezifikationen folgt. In den Teilaufgaben 6.2.2 und 6.2.3 soll das Skript noch erweitert werden. Füge das Gesamtskript aller drei Teilaufgaben in die Lösungsdatei ein.

Die Aufgabe des Skriptes ist es, einen Text von der Standardeingabe zu lesen und die Häufigkeit des Auftretens der verschiedenen Wörter zu zählen. Die Ausgabe soll dabei so formatiert sein, dass in jeder Zeile der Ausgabe die Häufigkeit eines Wortes auf 5 Zeichen Breite rechtsbündig formatiert werden soll, dann ein Leerzeichen folgt, und dann das Wort ausgegeben wird.

Beispiel

```
> python occurrence.py < /share/Courses/CG2011/blatt06/gpl-3.0.txt
 11: free
 11: OR
   .
   .
```

Hinweise

- Benutze die Funktionen `sys.stdin.read()` und `string.split()`.
- Speichere die Häufigkeit der einzelnen Wörter in einem Dictionary.
- 6.2.2 (1,5 Punkte) Erweitere das in Aufgabe 6.2.1 geschriebene Skript so, dass die Ausgabe nach den Häufigkeiten sortiert ist, und dass die am häufigsten auftretenden Wörter am Ende ausgegeben werden.

Hinweis Verwende die Funktion `sorted`. Wie man die Funktion genau verwendet solltest Du im Internet nachlesen. Die Funktion `help` hilft in diesem Fall leider nicht sehr.

- 6.2.3 (1,5 Punkte) Erweitere das in Aufgabe 6.2.1 geschriebene Skript so, dass es die Option `-m` bzw. `--min` versteht. Der Wert dieser Option soll festlegen, wie häufig ein Wort in der Eingabe mindestens auftreten muss, damit seine Häufigkeit ausgegeben wird.

Hinweis Verwende das Modul `optparse`.

Beispiel

```
> python occurrence.py --min=170 < /share/Courses/CG2011/blatt06/gpl-3.0.txt
 174 to
 208 of
 309 the
```

- 6.2.4 (1 Punkt) Verwende das Skript aus den letzten Aufgaben im Zusammenhang mit passenden Unixbefehlen, um festzustellen, welches die fünf häufigst auftauchenden Dateiname im Verzeichnis `/usr/share` und dessen Unterverzeichnissen sind. Wichtig dabei ist nur der Dateiname selbst, nicht die Namen der Unterverzeichnisse. Füge die Dateinamen und den dafür notwendigen Befehl in die Lösungsdatei ein.

Hinweis Verwende den Befehl `ls` mit den richtigen Optionen, nicht den Befehl `find`.