Übungen zu Computergrundlagen WS 2013/2014

Übungsblatt 11: Python

9.1.2014

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist
  - **Dienstag, 14.1.2014, 10:00** für die Übungsgruppen am Donnerstag und Freitag
  - **Mittwoch, 15.1.2014, 10:00** für die Übungsgruppen am Montag und Dienstag
- Schickt die Lösungen bitte per Email an Euren Tutor.

Aufgabe 11.1: Funktionen, Rekursionen und Schleifen (6 Punkte)

- **11.1.1** Die Sequenz der Fibonacci-Zahlen ist wie folgt definiert:

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

  Schreibt eine Python-Funktion `fib1(n)`, die die n-te Fibonacci-Zahl wie in Gleichung 1 berechnet. (1 Punkt)

  Hinweis: Das Programm `/group/cgl/2013/11/pow.py` könnt Ihr als Vorlage für Euer Programm benutzen. Es enthält die Funktion `pow`, die die Exponentialfunktion x^n berechnet, die mathematisch wie folgt definiert werden kann:

$$\text{pow}(x, n) = \begin{cases} 1 & \text{falls } n = 0 \\ x \cdot \text{pow}(x, n-1) & \text{sonst} \end{cases} \tag{2}$$

- **11.1.2** Schreibt eine Funktion `fib2`, die die Fibonacci-Zahlen mit Hilfe einer Schleife berechnet, also ohne den rekursiven Aufruf der Funktion. (2 Punkte)

  Hinweis: Berechnet zunächst auf dem Papier von Hand die ersten paar Fibonacci-Zahlen. Das macht es einfacher, zu verstehen, wie man die Schleife implementieren kann.

- **11.1.3** Eine andere, aber äquivalente, Definition der Fibonacci-Zahlen sieht so aus:

$$\text{fib}_{a,b}(n) = \begin{cases} a & \text{falls } n = 0 \\ b & \text{falls } n = 1 \\ \text{fib}_{b,(a+b)}(n-1) & \text{sonst} \end{cases} \tag{3}$$

$$\text{fib}(n) = \text{fib}_{1,1}(n) \tag{4}$$

  Schreibt eine Python-Funktion `fib3`, die die Fibonacci-Zahlen wie in Gleichung 4 berechnet. (2 Punkte)

  Hinweis: Am besten definiert Ihr die Indizes a und b in Python als optionale Parameter:

```
def fib3(n, a=1, b=1):
  # Hier kommt der Code!
```

- **11.1.4** Berechnet mit Hilfe der Funktionen `fib1`, `fib2` und `fib3` aus den vorigen Aufgaben die Fibonacci-Zahl für $n = 35$. Was fällt Euch bei den Laufzeiten auf? Wieso sind sie so unterschiedlich? (1 Punkt)

  Hinweis: Zur Laufzeitmessung könnt Ihr den Shellbefehl `time` wie folgt verwenden (in der Shell, *nicht* in Python!):

```
> time python fib.py
121393

real    0m0.080s
user    0m0.070s
sys     0m0.000s
```

Aufgabe 11.2: Python: Dictionaries, Strings und Module (4 Punkte)

In dieser Aufgabe sollt Ihr ein Pythonskript schreiben, das zählt, wie häufig verschiedene Buchstaben in einer Datei vorkommen. Das Skript soll den folgenden Spezifikationen folgen:

- Es liest eine Datei und zählt, wie häufig die einzelnen Buchstaben in der Datei vorkommen. Dabei soll es nur die alphanumerischen Zeichen zählen (also alle Buchstaben und Zahlen), nicht jedoch Satzzeichen und Leerzeichen. Groß- und Kleinbuchstaben sollen zusammengefasst werden, sie sollen also *nicht* getrennt gezählt werden. Die Ausgabe sollte ungefähr so aussehen. (2 Punkte)
- Die benötigten Dateinamen liest es über die Kommandozeile ein. Dabei soll es auch in der Lage sein, die Buchstabenhäufigkeiten in mehreren Dateien zu bestimmen. (1 Punkt)
- Die Ausgabe soll absteigend nach der Häufigkeit der Buchstaben sortiert sein. (1 Punkt)

Beispiel

```
> python letters.py file1.txt file2.txt
"a": 37
"o": 25
.
.
.
```

Hinweise:

- Um festzustellen, ob ein Zeichen alphanumerisch ist, kann man die Funktion `isalnum` wie folgt verwenden:

```
    s = 'a'
    if s.isalnum(): print "Alphanumerisch!"
```

- Als Beispieldatei könnt Ihr die Datei `/group/cgl/2013/11/gpl-3.0.txt` verwenden.