

Einführung in die wissenschaftliche Programmierung

Übungsblatt 4

1.) Ratespiel

Schreiben Sie ein Programm, mit welchem der Benutzer eine Zufallszahl erraten kann. Dies soll so ablaufen: Das Programm erzeugt eine Zufallszahl (z.B. natürliche Zahlen zwischen 0 und 1000) und der Benutzer kann seine Vermutung eingeben. Das Programm gibt dann entweder “zu groß”, “zu klein” oder “richtig” zurück. Verwenden Sie z.B. die `raw_input()` Funktion.

2.) Telefonbuch

Implementieren Sie in Python ein interaktives Telefonbuch mit den Funktionen “neuer Eintrag”, “Eintrag suchen” und “Alles anzeigen”. Verwenden Sie eine geeignete Datenstruktur um die Einträge (d.h. Name und Nummer) zu speichern. Wenn man das Python Programm startet, soll ein (einfaches, text-basiertes) Menü erscheinen, z.B.

- (1) `new entry`
- (2) `find entry`
- (3) `list entries`
- (4) `quit`

Je nach Auswahl des Benutzers, soll dann die gewünschte Funktion durchgeführt werden. Achten Sie im Menü auf ungültige Eingaben und reagieren Sie entsprechend darauf.

3.) Caesar Verschlüsselung

Auf Blatt 2 haben Sie die Caesar Verschlüsselung kennengelernt. Die dadurch verschlüsselten Texte können leicht entschlüsselt werden. Schreiben Sie ein Python Programm, welches einen Text aus einer Datei einliest und feststellt, welcher

“shift-Wert” verwendet wurde. Gehen Sie davon aus, dass es sich um einen Text in Englischer Sprache handelt und machen Sie sich das zunutze. Es gibt dafür verschiedene Möglichkeiten!

Benutzen Sie brauchbare Funktionen der Python Standard Library. Es hat sich dabei die `find` Methode von Strings als nützlich erwiesen: wenn Sie einen String `text = 'abcdef'` haben, können Sie darin nach einem Suchwort `'bc'` mit `text.find('bc')` suchen. Weiters können sie z.B. mit `a = 3` und `text.find('bc', a)` bestimmen, dass erst ab der Stelle 3 gesucht werden soll. Genaueres zu dieser Methode finden Sie in der Dokumentation der Python Standard Library unter

<http://docs.python.org/library/>

Auf der Webseite zur Veranstaltung finden Sie drei verschlüsselte Texte. Testen Sie ihr Programm mit diesen Texten.

4.) Cosmological Parameter Estimation

The cosmic microwave background (CMB) radiation has been emitted just after the Big Bang (around 400,000 years later) and has played an important role in the acceptance of the Big Bang Theory as standard. Since the statistical properties of the CMB depend on the parameters of the cosmological model, the CMB can be used to infer information about these parameters.



For a particular cosmological model, one is interested, e.g., in the marginal distributions of the parameters. If the likelihood \mathcal{L} was given in an explicit form, integrating in all but one dimension would provide the marginal distribution for the remaining one; however, as the likelihood function is not available, typically Markov Chain Monte Carlo (MCMC) methods are used instead, randomly drawing points with approximately the same distribution.

Seven chains have been run in parallel, leading to the files `chain0.dat`, `...`, `chain6.dat`. Each line of the files contains data for one point of the corresponding chain. The last column is the multiplicity of that point. As soon as the chains have converged, the last column contains the entry 0 which can occur only once per file. The aim is to obtain all points after convergence in one file with the filename `distilled.dat`, i.e., all lines of the seven files starting with the line after the one containing the 0.

(D. Pflüger *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut, February 2010.)

5.) Scrabble™ Hilfe

Schreiben Sie ein Python-Skript zur Unterstützung von Scrabble™-Spielern:
Nach der Eingabe von bis zu acht beliebigen Buchstaben sollen alle gültigen Wörter die aus diesen Buchstaben gebildet werden können in der Reihenfolge ihrer Wortwertigkeit ausgegeben werden.

Um die Gültigkeit von Worten zu prüfen verwenden Sie eine geeignete Wortliste (z.B. Google-Suche nach *Wortlisten*: Seite der Uni Leipzig mit Listen der häufigsten Worte in verschiedenen Sprachen). Die Wortwertigkeit berechnet sich aus der Summe der den einzelnen Buchstaben zugeordneten Werte, dies sind bei Scrabble™:

```
wert = {"A":1, "B":3, "C":4, "D":1, "E":1, "F":4, "G":2, "H":2,  
        "I":1, "J":6, "K":4, "L":2, "M":3, "N":1, "O":2, "P":4,  
        "Q":10, "R":1, "S":1, "T":1, "U":1, "V":6, "W":3, "X":8,  
        "Y":10, "Z":3}
```