Übungen zu Computergrundlagen WS 2019/2020

Übungsblatt 10: Python II

10. Januar 2019

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist **Freitag, 17.01.2020, 11:00 Uhr**
- Schickt die Lösungen bitte per Email an Euren Tutor:
  - Montag 14:00–15:30: Moritz Schumacher (`mschumacher@icp.uni-stuttgart.de`)
  - Dienstag 9:45–11:15: Samuel Tovey (`stovey@icp.uni-stuttgart.de`)
  - Dienstag 15:45–17:15: Philipp Stärk (`pstaerk@icp.uni-stuttgart.de`)
  - Mittwoch 15:45–17:15: Marco Brückner (`mbrueckner@icp.uni-stuttgart.de`)
  - **geändert:** Donnerstag 9:45–11:15: Ingo Tischler (`itischler@icp.uni-stuttgart.de`)
- Die Übungen sollen von Gruppen von jeweils *zwei* (nur in Ausnahmefällen drei) Leuten bearbeitet werden. Bitte gebt *nur eine Lösung pro Gruppe* ab und nennt in eurer Abgabe alle Mitglieder eurer Gruppe!
- Als Lösung der Aufgabe soll ein einziges Python-Skript erstellt werden, welche ihr dann per E-Mail an euren Tutor schickt.

Aufgabe 10.1: List Comprehensions, `map` und `filter` (4 Punkte)

In der Datei `/group/cgl/2019/10/einkaufsliste.txt` steht der Einkaufszettel für eine große Apfelkuchenparty. In jeder Zeile steht, durch Strichpunkt getrennt, ein Artikel, die benötigte Anzahl, der Stückpreis, sowie das Geschäft und ggf. die Abteilung in dem jeweiligen Geschäft, wo der Artikel zu finden ist.

- Lest diese Datei in eine verschachtelte Python-Liste ein. Diese sollte so aussehen:

  `[['Mehl', '3', '0.79', 'Supermarkt', 'Backzutaten'], ['Butter', ...], ...]`.

  Dabei sollten keine Zeilenumbrüche mehr an den Zeichenketten hängen. (1 Punkt)
- Filtert aus dieser Liste nur die Artikel heraus, die im Supermarkt gekauft werden sollen. Macht das einmal mit einer List Commprehension und einmal mit `filter()`. (1 Punkt)
- Sortiert die Supermarkt-Artikel nach der Abteilung, in der sie zu finden sind. Verwendet dazu `sorted()` mit einer `lambda`-Funktion. (1 Punkt).
- Berechnet, wie viel Geld im Getränkemarkt ausgegeben wird. Verwendet dazu eine List Comprehension. Berechnet, wie viel Geld in allen Geschäften zusammen ausgegeben wird. Verwendet dazu `map` und eine `lambda`-Funktion. (1 Punkt)

Hinweis: Die Summe aller Elemente einer Liste lässt sich mit `sum` berechnen.

Aufgabe 10.2: Kommandozeilenargumente parsen (3 Punkte)

In der Datei `/group/cgl/2019/10/change_base.py` sind Pythonfunktionen, die Zahlen in ein anderes Stellenwertsystem umrechnen. Eure Aufgabe ist es, mit Hilfe von `argparse` ein Skript zu schreiben, dem man auf der Kommandozeile das Quell- und Ziel-System angeben kann, außerdem ein Flag, ob ein Zwischenergebnis in der Basis 10 ausgegeben werden soll, und natürlich ein odere mehrere umzurechnende Zahlen. Schreibt Kommentare um zu erklären, was welche Option von `argparse` jeweils bewirkt. (3 Punkte)

Wenn man dieses Skript mit dem Argument `--help` aufruft, sollte das ungefähr so aussehen:

```
usage: solution.py [-h] [-f SRC] [-t DEST] [--intermediate] N [N ...]

Convert integers between different bases

positional arguments:
  N                     integer(s) to convert

optional arguments:
  -h, --help            show this help message and exit
  -f SRC, --from SRC    base of input number
  -t DEST, --to DEST    base of output number
  --intermediate        Whether to display the intermediate base-10 number
                        (false by default)
```

Hinweis: Bitte verändert *change_base.py* nicht und kopiert seinen Inhalt auch nicht in euer Skript; benutzt stattdessen **`import`**, um die benötigten Funktionen direkt von dort zu benutzen.

Aufgabe 10.3: NumPy und Matplotlib (3 Punkte)

10.3.1 Generiert Stützstellen im Intervall $[-2\pi,2\pi]$ und wertet an diesen die Sinus- und die Kosinusfunktion aus. Die Anzahl der Stützstellen sollte groß genug sein, dass in der nächsten Teilaufgabe glatte Kurven entstehen. (1 Punkt)

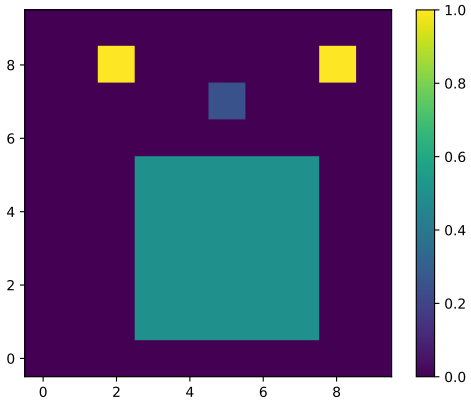
10.3.2 Plottet die beiden Funktionen in einem einzigen Diagramm. Vergesst nicht, die Legende, einen Titel und die Achsenbeschriftungen hinzuzufügen. (1 Punkt)

10.3.3 Erstellt ein NumPy-Array namens `data` mit 10×10 Nullen und verwendet Array-Zugriffe und -Slicing, um Werte zu setzen. Der erste Index ist die x-Koordinate und der zweite die y-Koordinate. (1 Punkt)

Am Ende soll

```
pyplot.imshow(data.T, origin='lower')
pyplot.colorbar()
pyplot.show()
```

folgendes Bild erzeugen:



Hinweis: Benutzt die Dokumentationen von NumPy und Matplotlib, um herauszufinden, wie die Funktionen heißen, die ihr braucht.