WIF

DVM

Python 101, Testklausur 2024-06-17 (mit Exahm)

Prof. Dr. Johannes Busse, HAW LA, busse@haw-landshut.de

- **Name, Vorname** nicht hier, sondern **auf dem letzten Blatt eintragen**
- **Anfangsbuchstaben vom Familiennamen** am **rechten Rand einkreisen**
- **Studiengang** am **linken Rand einkreisen**
- Falls Sie mehr Schreibraum benötigen: Nutzen Sie **nicht die Rückseite**, sondern die jeweils **gegenüberliegende Seite** einer Aufgabe.
- Verwenden Sie einen **dokumentenechten Stift** ungleich rot.
  Grafische Lösungen etc. dürfen Sie gerne auch mit Bleistift durchführen.
- Bitte lassen Sie den **Korrektur-Rand** rechts **frei.**
- **Schreiben Sie deutlich**. Unterstützen Sie uns dabei, Ihre Lösung zu erkennen.

Diese Klausur hat 40 Pkte (=100%), plus ein paar wenige Sonderpunkte.
Unverbindlicher Notenspiegel: ab 18 Punkte = Note 4, ab 24=3, ab 30=2, ab 36=1.

Inhaltsverzeichnis

Erstkorrektur J.Busse, Summe Pkte / Note:

ggf. Zweitkorrektur:

WICHTIG: Dieses Jupyter Notebook ist in der Klausur python-101 lediglich ein erlaubtes Hilfsmittel unter vielen anderen Hilfsmitteln, ähnlich einem anonym benutzbaren Taschenrechner. Zwar wird dieses Notebook technikbedingt zeitweise gespeichert, aber es gibt keine Verbindung zu Ihrer Person. Insbesondere können ihre Eingaben im Notebook vom Dozenten nicht eingesehen und auch nicht bewertet werden. Also:

Übertragen Sie alle Ergebnisse aus dem Notebook auf Papier.
Es wird ausschließlich die Papier-Version der Klausur bewertet.

← Für Notizen, Nebenrechnungen bitte ggf. auch die gegenüberliegende Seite verwenden

A B C D E F G H IJ K L M N O PQ R S Sch T U VW XYZ

Operatoren

gegeben seien die folgenden Variablen:

```
l = [ 1, 2, 3 ]
hallo = "Hallo!"
s = { "a", "l", "o"  }
```

gesucht: Verwenden Sie obige Variablen sowie eine geeignete Programmierung, um die Tests auf Gleichheit Wahr zu machen. Beispiel:

```
"Hallo!Hallo!" == ...
# Lösung:
"Hallo!Hallo!" == 2 * hallo

True
```

Jetzt sind Sie dran:

```
"allo" == ...




'!ollaH' == ...




'H_a_l_l_o_!' == ...




{ "H", "!" } == ...




['H', 'a', 'l', 'l', 'o', '!'] == ...




['1', '2', '3'] == …
```

“Schnittliste” von Listen

gegeben: zwei Listen, z.B.

```
l1 = [ 1, 2, 3, 2, 1 ]
l2 = [ 2, 3, 4, 5 ]
```

Die “Schnittliste” von zwei Listen sei definiert als alle Elemente von Liste 1, die auch in Liste 2 enthalten sind. Im Gegensatz zu einer Schnittmenge können die Elemente einer Liste mehrfach auftreten, und auch die Reihenfolge bleibt gleich.

```
def listen_schnittmenge(x, y):









    return ergebnis

listen_schnittmenge( l1, l2 ) == [2, 3, 2]
```

Dictionary: Werte addieren

gegeben:

- ein Dict von Würfelpunkten, z.B.

```
punkte = { "Anna": [ 2, 1, 6, 4 ], "Peter": [ 6, 3, 1, 2 ] }
```

gesucht:

- ein zweites Dict, das zu jedem Spieler die Summe der Punkte angibt.

```
... #









punkte_summe == {'Anna': 13, 'Peter': 12}
```

Dictionary: Gewinner ermitteln

gegeben:

- ein Dict von aufsummierten Würfelpunkten, z.B.

```
punkte = {'Anna': 7, 'Peter': 12, 'Charly': 15, 'Mary': 7, 'Tim': 13 }
```

gesucht:

- der Name des Gewinners, hier: des Spielers (m/w/d) mit den *meisten* Punkten

Teil 1

Lösung 1, vereinfachte idealisierte Annahme: Es gibt genau einen Spieler mit einer maximalen Punktzahl. Grundgerüst für einen Lösungsansatz mit konventioneller Schleife (aber jede andere Lösung ist auch ok, ggf. sogar besser):

```
m = 0
gewinner = None
for spieler, p in punkte.items():









gewinner == 'Charly'
```

Teil 2

Spielziel jetzt: Gewonnen haben alle Spieler mit einer minimalen Punktzahl.

Lösung 2, realistische Annahme: Mehrere Spieler können die gleiche minimale Punktzahl haben, wir erhalten also eine Menge von Gewinnern.

Lösungsidee: Minimale Punktzahl bestimmen, dann Menge der Spieler mit minimaler Punktzahl erzeugen. Aber man kann es natürlich auch ganz anders machen.

```









{'Anna', 'Mary'}
```

Dictionary invertieren

gegeben:

- ein Dictionary von Postleitzahlen, Beispiel:

```
d = { "Landshut": [ 84030, 84032, 84028 ], "Dingolfing": [ 84130 ] }
```

gesucht:

- ein inverses Dictionary, mit dem man bei einer gegebenen Postleitzahl den zugehörigen Ort nachschlagen kann.

```
d_invers = ...




d_invers == {84028: 'Landshut', 84130: 'Dingolfing', 84030: 'Landshut',
84032: 'Landshut'} # Reihenfolge ist egal
```

Textdatei auswerten

Der Metadaten-Abschnitt (der sog. Header) einer Email besteht aus einer Reihe von Attribut-Wert-Paaren.

In einer Text-Datei kann man den Header zeilenweise so notieren, Beispiel (vereinfacht):

```
header = """Date: Thu, 13 Jun 2024
User-Agent: Mozilla Thunderbird
Content-Language: de-DE
To: Johannes Busse <busse@haw-irgendwo.de>
Subject: test 09.29
"""
```

Beobachtungen:

- In jeder Zeile kommt `": "` genau ein mal vor.
- Jedes Attribut kommt höchstens ein mal vor.

Aufgabe Teil 1

gegeben:

- eine einzelne Zeile des Headers; Beispiel:

```
z2 = "Content-Language: de-DE"
```

gesucht:

- solch eine Zeile als eine Liste `[ <Attribut>, <Wert> ]`

```


z2_liste == [ "Content-Language", "de-DE" ]
```

Aufgabe Teil 2

Voraussetzung:

- eine Lösung zu Teil 1: Python Code, der einen Attribut-Wert-String `aw_string` (wie z.B. `z2`) in eine Liste `[ <Attribut>, <Wert> ]` verwandelt.

gesucht:

- die Lösung zu Teil 1 in eine Funktion `aw(aw_string)` eingepackt
- Rückgabewert der Funktion ist **nicht** die Liste `[ <Attribut>, <Wert> ]`, sondern der erste und der zweite Wert (das Attribut und sein Wert) aus dieser Liste

```
def aw(aw_string):
    ... #









    return attribut, wert

z2 = "Content-Language: de-DE"
aw(z2) == ('Content-Language', 'de-DE')
```

Aufgabe Teil 3

(Diese Teilaufgabe ist für die Klausur zu umfangreich, aber für die Klausurvorbereitung zuhause sollte es gut machbar sein, siehe die Kompetenzanforderungen unter https://www.jbusse.de/jvdp-jb/python-101-kompetenzen-a1.html)

gegeben:

- der komplette E-Mail Header als String

gesucht:

- der Header als ein Dictionary `header_dict`

```
header = """Date: Thu, 13 Jun 2024
User-Agent: Mozilla Thunderbird
Content-Language: de-DE
To: Johannes Busse <busse@haw-irgendwo.de>
Subject: test 09.29"""
```

(siehe die Online-Version dieses Aufgabe unter https://www.jbusse.de/python-101/testklausur_2024-06-17_AUTO.html#aufgabe-teil-3

Persönliche Angaben

Familienname:

Vorname:

Matrikelnummer:

Studiengang: