

Aufgabenblatt 5, Abgabetermin 15.6.2020

Lösen Sie drei Aufgaben. Sie können maximal 15 Punkte erreichen.

Aufgabe 28: Vergrößerung

5 Punkte

Schreiben Sie ein Programm, das eine Eingabe von standard input liest und „vergrößert“, indem jedes Zeichen `x` durch `xx` ersetzt. Testen Sie das Programm, indem Sie das Haus des Nikolaus vergrößern. Dazu müssen Sie das Programm für das Haus des Nikolaus so umschreiben, dass es die Eingabe von der Kommandozeile liest. Das folgende Beispiel zeigt ein Haus vom Nikolaus der Größe 3 und seine Vergrößerung.

`python3 nikolaus.py 3`

```
*
* *
*****
* * *
*****
```

`python3 nikolaus.py 3 | python3 blowup.py`

```
**
**
** **
** **
*****
*****
** ** **
** ** **
*****
*****
```

Aufgabe 29: Lottozahlen

5 Punkte

Die Datei `lottozahlen_archiv.csv` enthält die Ergebnisse der Lotto-Ziehungen seit 1955. Jede Zeile der Datei enthält die Daten zu einer Ziehung. Welche Zahl ist am häufigsten und welche am seltensten gezogen worden?

Aufgabe 30: Magische Quadrate erzeugen und testen

5 Punkte

Bauen Sie die Programme aus den Aufgaben 23 und 24 so um, dass das eine Programm (Aufgabe 24) das erzeugte magische Quadrat ausgibt, und das andere Programm (Aufgabe 23) diese Ausgabe einlesen kann und testet, ob die eingelesenen Zahlen ein magisches Quadrat ergeben. Testen Sie dieses Programm auch mit Eingaben, die keine magischen Quadrate sind.

Aufgabe 31: word count

5 Punkte

Der Linux-Befehl `wc Dateiname` gibt die Anzahl der newlines `\n`, der Wörter und der Zeichen in der Datei `Dateiname` zurück. Zum Beispiel:

```
$ wc Jena-Wetterdaten.txt
69986  877570 9346093 Jena-Wetterdaten.txt
```

Also enthält die Datei `Jena-Wetterdaten.txt` 69986-mal `\n`, 877570 Wörter und 9346093 Zeichen. (Ein *Wort* ist eine Folge von Zeichen (also ein String) mit Länge ≥ 1 , die kein Leerzeichen, Zeilenumbruch etc. enthält. Der String `'a bc\n(1 2d\n'` besteht aus 4 Wörtern und 12 Zeichen.)

Schreiben Sie ein Python-Programm, das einen Dateinamen von der Kommandozeile liest und die gleiche Ausgabe wie `wc Dateiname` produziert.

(Das Modul `string` hat eine Variable `string.whitespace`. Sie ist ein String aus allen Leerzeichen etc. `' \t\n\r\x0b\x0c'`.)

Aufgabe 32: Entfernung zum nächsten Flughafen

5 Punkte

Die Datei `airports.csv` enthält zeilenweise Daten von allen Flughäfen, zum Beispiel den Namen und die Lage in Längen- und Breitengrad. Sie stammt von [openflights.org](https://openflights.org/data.html) von der Seite <https://openflights.org/data.html>. Dort finden Sie auch Informationen zu den Einträgen.

Schreiben Sie ein Programm, das eine Position auf der Erdkugel als Längen- und Breitengrad (`float`) einliest und den nächsten Flughafen und dessen Entfernung ausgibt.

Um die Entfernung zwischen zwei Punkten auf einer Kugeloberfläche zu berechnen, können Sie die Formel für die Orthodrome (siehe <https://de.wikipedia.org/wiki/Orthodrome>) benutzen. Beachten Sie, dass Pythons trigonometrische Funktionen das Bogenmaß benutzen. Um Gradzahlen in das Bogenmaß umzurechnen, können Sie die Funktion `math.radians()` benutzen.

Korrelation von Messreihen

Wir haben zwei gleichlange Messreihen x_1, x_2, \dots, x_n und y_1, y_2, \dots, y_n mit Messwerten (z.B. ist x_i die Anzahl der Stunden mit Sonnenschein im Jahr i und y_i ist die Niederschlagsmenge im Jahr i). Sei \bar{x} der Durchschnitt aller x_i und \bar{y} der Durchschnitt aller y_i (also die durchschnittliche Jahressonnenscheindauer bzw. Jahresniederschlagsmenge im Messzeitraum). Es gibt eine Korrelation zwischen den beiden Messreihen, wenn es häufig (d.h. z.B. für mehr als $\frac{2}{3}$ aller i) vorkommt, dass

- $x_i < \bar{x}$ und $y_i < \bar{y}$, oder $x_i \geq \bar{x}$ und $y_i \geq \bar{y}$ (positive Korrelation)
(„wenig Sonne, wenig Regen, und viel Sonne, viel Regen“), oder
 - $x_i < \bar{x}$ und $y_i \geq \bar{y}$, oder $x_i \geq \bar{x}$ und $y_i < \bar{y}$ (negative Korrelation)
(„wenig Sonne, viel Regen, und viel Sonne, wenig Regen“).
-

Aufgabe 33: Jahressummen von 2 Spalten der DWD-Datei

5 Punkte

Mit dieser und der folgenden Aufgabe sollen Programme geschrieben werden, mit denen man einen Zusammenhang (Korrelation) zwischen zwei Wetterereignissen feststellen kann – zum Beispiel zwischen der Jahressonnenscheindauer und der Jahresregenmenge.

In dieser Aufgabe sollen die Jahreswerte ermittelt werden.

1. Schreiben Sie ein Programm `doppelfilter_jahr.py`, das zwei `int`-Werte a und b von der Kommandozeile liest, und auf standard output für jede Zeile der eingelesenen DWD-Datei das Jahr und die Werte aus Spalte a und Spalte b ausgibt. (Dieses Programm ist eine Variante von `filter-jahr-tmk.py`.)
2. Schreiben Sie ein Programm `jahres_summen.py`, das für jedes Jahr die Summen der Spalten ausgibt (ohne das Jahr davorzuschreiben). (Dieses Programm ist eine Variante von `jahres-mittel.py`.)

Beide Programme hintereinander sollen z.B. etwa folgende Ausgabe geben.

```
$ python3 doppelfilter-jahr.py 15 16 < Jena-Wetterdaten.txt | python3 jahres-summen.py
4940.9 1315.1
4882.4 1969.1
4842.1 1894.4
4717.4 1760.3
4784.3 2137.2
3821.5 1118.6
...
```

Achten Sie darauf, dass für Jahre, bei denen in einer Spalte viele Werte fehlen, keine der beiden Summen ausgegeben wird.

Aufgabe 34: Korrelation

5 Punkte

Schreiben Sie ein Programm, das von standard input Zeilen einliest, die jeweils aus zwei `float`-Werten bestehen (wie die Ausgabe in Aufgabe 33). Der erste Wert soll als Wert der einen Messreihe (x -Wert) und der zweite als zugehöriger Wert einer anderen Messreihe (y -Wert) aufgefasst werden. Das Programm soll feststellen, ob es eine Korrelation zwischen den beiden Messreihen gibt.

Nutzen Sie die Programme aus Aufgaben 33 und 34, um Beispiele für Messwerte aus der DWD-Datei zu finden, zwischen denen es eine Korrelation gibt bzw. zwischen denen es keine Korrelation gibt.

Aufgabe 35: Bauernregeln

5 Punkte

Es gibt die Bauernregel: „Wenn’s am Siebenschläfer regnet, sind wir sieben Wochen mit Regen gesegnet.“ Der Siebenschläfertag ist am 27. Juni. Lässt sich diese Bauernregel durch die DWD-Wetterdaten für Jena (Datei `Jena-Wetterdaten.txt`) bestätigen?