

Junioraufgabe 1: Parallelen

Team-ID: #####

Team-Name: ###

Bearbeiter/-innen dieser Aufgabe:
Snocember

23. November 2019

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	2
Quellcode.....	4

Lösungsidee

Mithilfe des Programms soll Martins Aussage, die in der Aufgabe genannt wird, bestätigt werden. Er meint, dass wenn man von einem beliebigen Wort der ersten Hälfte immer die Anzahl n der Buchstaben in diesem Wort zum nächsten Wort springt, immer am Ende, wenn man nicht mehr „weeterspringen“ kann, beim selben Wort endet.

Also testet das Programm jedes Wort in der ersten Hälfte als Startwort aus und berechnet das Endwort. Wie vom Martin beschrieben, springt man immer die Buchstabenanzahl n des Wortes weiter zum nächsten Wort, bis es bei der nächsten Zahl kein Wort mehr gibt. Es wird nach jedem Durchgang das erste Endwort mit dem aktuellen Endwort verglichen. Sollten beide nicht miteinander übereinstimmen, wird ausgegeben, dass Martin Unrecht hat. Tritt nach jedem Durchgang eine Übereinstimmung ein, wird ausgegeben, dass Martin Recht hat.

Umsetzung

Die Lösungsidee wird in Python 3.7.0 implementiert. Es wurde MacOS Mojave mit Terminal.app genutzt.

Die heruntergeladene Textdatei ‚parallelen.txt‘ wird geladen, alle Sonderzeichen und Satzzeichen werden entfernt und in eine Liste umgewandelt. Beginnend mit einer For-Schleife wird die Buchstabenanzahl n des Anfangswortes, das aus dem Anfangsindex und einem Counter berechnet wird, gezählt und n Listeneinträge weitergesprungen. Nach diesem Sprung (und nach jedem weiteren Springen) wird das Wort und die Buchstabenanzahl ausgegeben. Dann wird in eine While-Schleife übergegangen, wo wiederum die Anzahl der Buchstaben des Wortes gezählt und von dem Zweitwort, berechnet nach dem Anfangswort, weitergesprungen wird, bis ein IndexError auftritt, der abgefangen wird. In der Exception, die den Error abfängt, wird das letzte Endwort mit dem letzten Wort vor dem IndexError verglichen, bei Ungleichheiten werden beide Schleifen abgebrochen und ausgegeben, dass Martin Unrecht hat. Wenn beide Endwörter übereinstimmen, wird die While-Schleife gebrochen, dem Counter der Wert 1 plusgeschrieben, womit jedes Mal das Anfangswort berechnet wird, und die For-Schleife beginnt erneut. Endet die For-Schleife, die alle Wörter der ersten Hälfte als Anfangswort nimmt, ohne einen IndexError, wird ausgegeben, dass Martin Recht hat.

Beispiele

Das Programm wird abgerufen.

```
$ python3 ./parallelen.py
```



EINGABE (Original vom der BWINF-Website, bereits hier vom Programm bearbeitet, sodass keine Satz- und Sonderzeichen vorkommen.)

```
['Es', 'gingen', 'zwei', 'Parallelen', 'ins', 'Endlose', 'hinaus', 'zwei', 'kerzengerade', 'Seelen',
'und', 'aus', 'solidem', 'Haus', 'Sie', 'wollten', 'sich', 'nicht', 'schneiden', 'bis', 'an', 'ihr',
'seliges', 'Grab', 'Das', 'war', 'nun', 'einmal', 'der', 'beiden', 'geheimer', 'Stolz', 'und',
'Stab', 'Doch', 'als', 'sie', 'zehn', 'Lichtjahre', 'gewandert', 'neben', 'sich', 'hin', 'da',
'wards', 'dem', 'einsamen', 'Paare', 'nicht', 'irdisch', 'mehr', 'zu', 'Sinn', 'Warn', 'sie',
'noch', 'Parallelen', 'Sie', 'wußstens', 'selber', 'nicht', 'sie', 'flossen', 'nur', 'wie', 'zwei',
'Seelen', 'zusammen', 'durch', 'ewiges', 'Licht', 'Das', 'ewige', 'Licht', 'durchdrang', 'sie',
'da', 'wurden', 'sie', 'eins', 'in', 'ihm', 'die', 'Ewigkeit', 'verschlang', 'sie', 'als', 'wie',
'zwei', 'Seraphim']
```

Die Hälfte des Gedichts endet, wie vorgeschrieben, bei: 'Doch als sie zehn Lichtjahre gewandert neben sich hin,'.

AUSWERTUNG

```
Es(2) -->zwei(4) -->hinaus(6) -->solidem(7) -->bis(3) -->seliges(7) -->beiden(6) -->als(3) -
>Lichtjahre(10) -->nicht(5) -->Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -
>ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
gingen(6) -->zwei(4) -->aus(3) -->Sie(3) -->nicht(5) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10)
-->nicht(5) -->Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -
>verschlang(10)
Endwort nach diesem Durchgang: verschlang
zwei(4) -->hinaus(6) -->solidem(7) -->bis(3) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -
>nicht(5) -->Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -
>verschlang(10)
Endwort nach diesem Durchgang: verschlang
Parallelen(10) -->Haus(4) -->nicht(5) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -
>Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
ins(3) -->zwei(4) -->aus(3) -->Sie(3) -->nicht(5) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -
>nicht(5) -->Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -
>verschlang(10)
Endwort nach diesem Durchgang: verschlang
Endlose(7) -->solidem(7) -->bis(3) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -
>Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
hinaus(6) -->solidem(7) -->bis(3) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -
>Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
zwei(4) -->aus(3) -->Sie(3) -->nicht(5) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5)
-->Warn(4) -->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
kerzengerade(12) -->an(2) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -->Warn(4) -
>Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
Seelen(6) -->wollten(7) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -->Warn(4) -
>Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
und(3) -->Haus(4) -->nicht(5) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -->Warn(4)
-->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
aus(3) -->Sie(3) -->nicht(5) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -->Warn(4)
-->Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
solidem(7) -->bis(3) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -->Warn(4) -
>Sie(3) -->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
Haus(4) -->nicht(5) -->seliges(7) -->beiden(6) -->als(3) -->Lichtjahre(10) -->nicht(5) -->Warn(4) -->Sie(3)
-->nicht(5) -->zwei(4) -->ewiges(6) -->sie(3) -->sie(3) -->ihm(3) -->verschlang(10)
Endwort nach diesem Durchgang: verschlang
```



```
neben(5) ->dem(3) ->nicht(5) ->Warn(4) ->Sie(3) ->nicht(5) ->zwei(4) ->ewiges(6) ->sie(3) ->sie(3)
->ihm(3) ->verschlang(10)
Endwort nach diesem Durchgang: verschlang
sich(4) ->dem(3) ->nicht(5) ->Warn(4) ->Sie(3) ->nicht(5) ->zwei(4) ->ewiges(6) ->sie(3) ->sie(3) -
>ihm(3) ->verschlang(10)
Endwort nach diesem Durchgang: verschlang
hin(3) ->dem(3) ->nicht(5) ->Warn(4) ->Sie(3) ->nicht(5) ->zwei(4) ->ewiges(6) ->sie(3) ->sie(3) -
>ihm(3) ->verschlang(10)
Endwort nach diesem Durchgang: verschlang
```

ERGEBNIS

Martin HAT RECHT!

Das Endwort nach jedem Durchgang (bis zur Hälfte des Gedichts) war: verschlang

Quellcode

```
#!/usr/bin/env python3
import sys                                # Für den Programmabbruch, falls keine Datei vorhanden
gedicht = []
zahlen = []
sonderz = ",.;:?!-\\n"                   # zu entfernende Sonderzeichen, die im Gedicht vorkommen

try:                                       # Die Sonder und Satzzeichen werden entfernt, danach das Gedicht in die Liste ,gedicht' eingetragen.
    file = open("./parallelen.txt")      # Die Datei wird geladen und...
    inhalt = file.readlines()            # ...eingelesen als ,inhalt'.
    groesse = len(inhalt)
    for i in range(0, groesse):
        text = inhalt[i].split()         # Der ,inhalt' wird wiederum aufgeteilt nach jedem Wort ...
        anzahl2 = len(text)
        for u in range(0, anzahl2):
            splittext = text[u]
            for x in sonderz:
                splittext = splittext.replace(x,"")    # Die Sonderzeichen werden entfernt.
            gedicht.append(splittext)        # ... und der Liste ,gedicht' hinzugefügt.
except FileNotFoundError:
    [...]
    sys.exit(0)                           # Programm wird abgebrochen, wenn kein Datei vorhanden ist.

gedicht.remove("")                        # Ein Leerzeichen wird entfernt
print(gedicht)

# Der Auswertungsteil
index = gedicht.index("hin")              # Das letzte Wort der ersten Hälfte ist „hin“.
nein = 0
endwort = ""
for x in range(0,index+1):
    lettern = len(gedicht[x])              # Buchstabenanzahl des Anfangswortes wird gezählt.
    print(gedicht[x]+"("+str(lettern)+") ", end=" ")    # Programmausgabe des Anfangswortes
    OK = True
    count = 0+lettern                      # Der Counter (der aktuelle Zeiger, bei welchem Wort man ist), wird auf die...
    while OK:                              # ...Buchstabenanzahl des Anfangswortes gesetzt, also wird „weitergehüpft“.
        Try:                               # Die while-Schleife mit try und except-Block
            newwort = gedicht[x+count]      # Das nächste Wort wird aus dem Index des Anfangswortes und dem Counter berechnet.
            lettern = len(gedicht[x+count]) # Buchstabenanzahl des nächsten Wortes wird gezählt.
            print("->"+newwort+"("+str(lettern)+")", end=" ") # Programmausgabe des nächsten Wortes
            count = count+lettern           # Dem Counter wird die Buchstabenanzahl des Wortes dazuaddiert (weiterhüpfen)
        except IndexError:                  # Falls das nächste Wort nicht mehr existiert und in der Leere ist:
            if x == 0:                      # Wenn es das erste Anfangswort ist, ...
                endwort = newwort           # ... wird dessen Endwort als Stamm-endwort festgelegt.
            if newwort != endwort:          # Wenn Ungleichheiten sind - >
```

```
    nein = 1                # Wichtig für das Endergebnis; wenn ,nein' != 0, dann hat Martin Unrecht
    print("ANDERES ENDWORT: "+newwort, end=" ")
    schleife = False        # Schleife wird durch die Änderung des Wertes abgebrochen
    print("Endwort nach diesem Durchgang: "+gelb+newwort+reset)
    break
if schleife == False:      # Wenn Ungleichheiten sind, dann wird der Auswertungsteil abgebrochen
    break
# Endergebnis
if nein == 0:              # ,nein' wird in der IndexException verändert
    print("Martin "+pink+"HAT RECHT!" +reset)
    print("Das Endwort nach jedem Durchgang (bis zur Hälfte des Gedichts) war: "+gelb+endwort+reset)
if nein != 0:
    print("Martin "+pink+"hat NICHT Recht!" +reset)
    print("Die Endwörter nach jedem Durchgang (bis zur Hälfte des Gedichts) war verschieden.")
```