

Junioraufgabe 1: Parallelen

24. November 2019

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	3
Quellcode.....	5

Lösungsidee

Die Idee ist, in einer Schleife jedes Wort anhand von Martins Aussage durchzuspielen. Die Wörter sollen dazu zuerst hintereinander in einer Liste abgespeichert werden. Folgende Probleme müssen dabei berücksichtigt werden:

- 1) Die Hälfte des Textes, also der Zeilen, ist nicht die Hälfte der Wörter. Am besten ist es deshalb bis zum letzten Wort der ersten Hälfte („hin“), d.h. bis genau vor dem ersten Wort der zweiten Hälfte („da“) zu überprüfen.
- 2) Return- und/oder Satzzeichen dürfen nicht bei der Berechnung der Wortlänge mitgezählt werden. Am besten ist es, sie vorher zu entfernen.

Die Endwörter die sich jeweils ergeben, sollen gespeichert werden. Nachdem alle Endwörter ermittelt worden sind, soll überprüft werden, ob alle Endwörter die selben sind. Zum Schluss soll das Ergebnis ausgegeben werden.

Umsetzung

Ich habe das Programm in der Programmiersprache Python in der Version 3.6.5 geschrieben.

Als erstes wird die Datei geöffnet und eingelesen, wobei schon direkt nach dem Einlesen die Return- und Satzzeichen gelöscht werden:

```
with open('parallelen.txt', encoding='utf-8') as file:
    for line in file:
        #Das Return am Zeilenende löschen:
        line = line.rstrip('\n')
        #De Satzzeichen löschen, weil die nicht zu den Wörtern zählen:
        line = line.replace('.', '')
```

Junioraufgabe 1: Parallelen

```
line = line.replace(',','')
line = line.replace(';','')
line = line.replace(':', '')
line = line.replace('?', '')
line = line.replace('-', '')
```

Danach wird jede eingelesene Zeile des Gedichts in eine Liste von Wörtern zerlegt und die Liste wird an die am Anfang leere Liste „wörter“ angehängt. Dabei werden auch die Leerzeichen entfernt:

```
wörter += line.split(' ')
```

Als nächstes werden die leeren Zeilen aus der Liste gelöscht, damit diese nicht als eigenes Wort gezählt werden:

```
wörter = [feld for feld in wörter if feld != '']
```

Als Trennung für die Hälfte, habe ich das erste Wort der zweiten Zeile vom dritten Absatz („da“) genommen, damit alle Wörter, einschließlich dem letzten Wort der ersten Hälfte des Gedichts, gemäß Martins Aussage in einer While-Schleife durchgespielt werden. Das Wort „da“ kommt bis zur Hälfte nicht noch einmal vor:

```
while wörter[anfangswort] != "da":
    #Das Wort von dem angefangen wird zu springen ausgeben
    print(str(anfangswort+1) + ". ", end="")
    #Das sichert den Wert von anfangswort in wort, damit man ihn verändern
    #kann, ohne die normale Reihenfolge kaputt zu machen:
    wort = anfangswort
```

In der ersten Schleife gibt es eine zweite While-Schleife, in der für das gerade ausgewählte Wort Martins Aussage durchgespielt wird:

```
while wort < anzahlwörter:
    #Das gibt das Wort, zu dem man gerade gesprungen ist,
    #und die Anzahl der Buchstaben des Wortes aus:
    print(wörter[word]+" (" +str(len(wörter[word]))+") ", end="")
    #Jetzt wird das Endwort abspeichern:
    letzteswort = wort
    #Dann wird zum nächsten Wort gesprungen:
    wort = wort + len(wörter[word])
```

Die Positionszahl vom letzten Wort, von dem aus zum Schluss nicht mehr weitergesprungen werden konnte, wird in der Liste „letztewörter“ abgespeichert, damit alle Wörter am Ende des Programms miteinander verglichen werden können:

```
#Das Schlusswort wird zu einer Liste hinzugefügt:
letztewörter.append(letzteswort)
#Eins hochzählen für das nächste Wort, bei dem das Durchspielen neu anfängt:
anfangswort += 1
```

Beinhaltet die Liste „letztewörter“ nur die gleichen Zahlen, also die gleichen Positionen der Endwörter in der Liste mit dem ganzen Text, dann hat Martin Recht. Ist aber eine oder sind mehrere Zahlen unterschiedlich, dann Stimmt Martins Aussage nicht. Das wird so überprüft:

```
#Es wird angenommen, dass alle letzten Wörter gleich sind:
toderf = True
#Jetzt werden alle Schlusswörter verglichen:
```

Junioraufgabe 1: Parallelen

```
for i in range(1, len(letztewörter)):
    if letztewörter[i] != letztewörter[0]:
        #Ein letztes Wort ist anders!
        toderf = False
```

Ganz zum Schluss wird dann noch das Ergebnis ausgegeben:

```
if toderf == False :
    #Sind sie alle Wörter gleich, dann hat Martin Recht:
    print('Martins hat nicht Recht!')
else:
    #Ist eines oder sind mehrere Wörter unterschiedlich, dann hat er Unrecht:
    print('Martin hat Recht!')
```

Beispiele

1. 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)

2. 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)

3. 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)

4. 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)

5. 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)

6. 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)

7. 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)

8. 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)

9. 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)

10. 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)

11. 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)

12. 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)

Junioraufgabe 1: Parallelen

13. 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)
14. 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)
15. 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)
16. 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)
17. sich (4) 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)
18. 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)
19. schneiden (9) einmal (6) Stab (4) zehn (4) sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
20. 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)
21. 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)
22. ihr (3) Das (3) einmal (6) Stab (4) zehn (4) sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
23. 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)
24. Grab (4) einmal (6) Stab (4) zehn (4) sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
25. Das (3) einmal (6) Stab (4) zehn (4) sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
26. war (3) der (3) Stolz (5) sie (3) gewandert (9) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
27. nun (3) 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)
28. einmal (6) Stab (4) zehn (4) sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
29. der (3) Stolz (5) sie (3) gewandert (9) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
30. 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)
31. geheimer (8) Lichtjahre (10) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
32. Stolz (5) sie (3) gewandert (9) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
33. und (3) als (3) Lichtjahre (10) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
34. Stab (4) zehn (4) sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)

Junioraufgabe 1: Parallelen

35. Doch (4) Lichtjahre (10) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
36. als (3) Lichtjahre (10) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
37. sie (3) gewandert (9) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
38. zehn (4) sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
39. Lichtjahre (10) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
40. gewandert (9) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
41. neben (5) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
42. sich (4) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)
43. hin (3) dem (3) nicht (5) Warn (4) Sie (3) nicht (5) zwei (4) ewiges (6) sie (3) sie (3) ihm (3) verschlang (10)

Martin hat Recht!

Quellcode

```
wörter = []
#Datei öffnen und Zeilen einlesen
with open('parallelen.txt', encoding='utf-8') as file:
    for line in file:
        #Return am Zeilenende löschen
        line = line.rstrip('\n')
        #Satzzeichen löschen, weil die nicht zu Wörtern zählen
        line = line.replace('.', '')
        line = line.replace(',', '')
        line = line.replace('; ', '')
        line = line.replace(':', '')
        line = line.replace('?', '')
        line = line.replace('-', '')
        #Die Leerzeichen werden entfernt
        #und die Wörter werden zu der Liste hinzugefügt:
        wörter += line.split(' ')
        #Die leeren Zeilen müssen gelöscht werden,
        #da sie sonst auch als Wörter gelten würden.
        wörter = [feld for feld in wörter if feld != '']

#Anzahl der Listenelemente herausfinden
anzahlwörter = len(wörter)
#Eine leere Liste anlegen, in die die Endwörter zum späteren Vergleich kommen.
letztewörter = []
#Das wort bei dem man anfängt zu springen auf Position 0 setzten.
anfangswort = 0
#Eine Schleife die sich wiederholt, bis man beim Wort 'da', dem ersten
#Wort im zweiten Abschnitt, ankommt:
while wörter[anfangswort] != "da":
    #Das Wort von dem angefangen wird zu springen ausgeben
    print(str(anfangswort+1) + ". ", end="")
    #Das sichert den Wert von anfangswort in wort, damit man ihn verändern
    #kann, ohne die normale Reihenfolge kaputt zu machen:
    wort = anfangswort

#Die Schleife wiederholt sich bis man nicht mehr
#weilerspringen kann, weil zu wenig Wörter da sind:
```

Junioraufgabe 1: Parallelen

```
while wort < anzahlwörter:
    #Das gibt das Wort, zu dem man gerade gesprungen ist,
    #und die Anzahl der Buchstaben des Wortes aus:
    print(wörter[word]+" (" +str(len(wörter[word]))+") ",end="")
    #Jetzt wird das Endwort abspeichern:
    letzteswort = wort
    #Dann wird zum nächsten Wort gesprungen:
    wort = wort + len(wörter[word])

print()
print()
#Das Schlusswort wird zu einer Liste hinzugefügt:
letztewörter.append(letzteswort)
#Eins hochzählen für das nächste Wort, bei dem das Durchspielen neu anfängt:
anfangswort += 1

#Es wird angenommen, dass alle letzten Wörter gleich sind:
toderf = True
#Jetzt werden alle Schlusswörter verglichen:
for i in range(1, len(letztewörter)):
    if letztewörter[i] != letztewörter[0]:
        #Ein letztes Wort ist anders!
        toderf = False

if toderf == False :
    #Sind sie alle Wörter gleich, dann hat Martin Recht:
    print('Martins hat nicht Recht!')
else:
    #Ist eines oder sind mehrere Wörter unterschiedlich, dann hat er Unrecht:
    print('Martin hat Recht!')
```