

Junioraufgabe 1: Reimerei

Team-ID: 00495

Team: SRZinfo4/1

Bearbeiter/-innen dieser Aufgabe:
Jakob Paridon

20. November 2022

Inhaltsverzeichnis

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

Lösungsidee

Die drei Kriterien werden nacheinander geprüft. Hierzu muss zuerst jede Vokalgruppen aller Wörter isoliert und gespeichert werden. Im Anschluss wird die maßgebliche Vokalgruppe ermittelt und darauf untersucht, ob sie und das, was ihr folgt, mindestens die Hälfte der Buchstaben enthält. Alle maßgeblichen Vokalgruppen, welche diese Kriterien erfüllen, werden gespeichert und miteinander abgeglichen, um alle sich reimenden Worte zu ermitteln.

Umsetzung

Das Programm wird mit `python junior1.py` [Name der Datei mit den zu untersuchenden Wörtern] bzw. `junior1.exe` [Name der Datei mit den zu untersuchenden Wörtern] gestartet.

Nach dem Einlesen der Wörter werden alle Indizes, an denen Vokalgruppen (inklusive ä, ü und ö) in einem Wort beginnen, gespeichert¹.

Im Anschluss wird der Beginn der maßgeblichen Vokalgruppen in einem Wort gespeichert, indem entweder kein Index (falls keine Gruppe gefunden wurde), der Index der einzigen gefunden Vokalgruppe oder der Index der vorletzten Vokalgruppe gespeichert wird².

Dann wird untersucht, ob die maßgebliche Vokalgruppe und die Buchstaben, welche ihr folgen, mindestens die Hälfte der Buchstaben des Wortes enthält. Ist dies der Fall, werden alle diese Buchstaben (die “Endgruppe”) gespeichert³.

Anschließend werden alle gespeicherten Endgruppen durchlaufen und miteinander verglichen, wobei alle Wörter, welche die gleiche Endgruppe besitzen, zusammen gespeichert werden, insofern das erste Wort nicht im zweiten oder das zweite Wort im ersten enthalten ist⁴.

Am Ende erfolgt die Ausgabe aller Wortpaare.

Beispiele

Die Testung erfolgte unter Artix Linux mit Python 3.10.7.

1.) reimerei0.txt

Lösung: bemühen, glühen

Lösung: Biene, Hygiene

Lösung: Biene, Schiene

Lösung: Hygiene, Schiene

Lösung: Knecht, Recht

2.) reimerei1.txt

Lösung: Bildnis, Wildnis

Lösung: Brote, Note

3.) reimerei2.txt

Lösung: Epsilon, Ypsilon

4.) reimerei3.txt

Lösung: Absender, Kalender

Lösung: Bahn, Zahn

Lösung: Bank, Dank

Lösung: Baum, Raum

Lösung: Bein, Wein

Lösung: Bier, Tier

Lösung: Bitte, Mitte

Lösung: Butter, Mutter

Lösung: Dame, Name

Lösung: Dezember, November

Lösung: Dezember, September

Lösung: Drucker, Zucker

Lösung: Durst, Wurst

Lösung: Fest, Test

Lösung: Feuer, Steuer

Lösung: Fisch, Tisch

Lösung: Flasche, Tasche

Lösung: Fluss, Fuß

Lösung: Glück, Stück

Lösung: Gleis, Kreis

Lösung: Gleis, Preis

Lösung: Gleis, Reis

Lösung: Gruppe, Suppe

Lösung: Hand, Land

Lösung: Hose, Rose
Lösung: Hund, Mund
Lösung: Kündigung, Reinigung
Lösung: Kanne, Panne
Lösung: Kasse, Klasse
Lösung: Kasse, Tasse
Lösung: Keller, Teller
Lösung: Kind, Rind
Lösung: Kind, Wind
Lösung: Klasse, Tasse
Lösung: Kopf, Topf
Lösung: Kreis, Preis
Lösung: Kunde, Stunde
Lösung: Lohn, Sohn
Lösung: Magen, Wagen
Lösung: November, September
Lösung: Platz, Satz
Lösung: Rind, Wind
Lösung: Rock, Stock
Lösung: Sache, Sprache
Lösung: Sekunde, Stunde

Quellcode

1.) Ermitteln der Startindizes aller Vokalgruppen in einem Wort

```
def getGroupIndices(word):
    vowels = "aeiouäüö"
    groupIndices = []
    groupIndex = -1
    wordIndex = 0
    while wordIndex < len(word):
        if word[wordIndex] in vowels:
            groupIndex = wordIndex
            while wordIndex < len(word) and word[wordIndex] in vowels:
                wordIndex += 1
            groupIndices.append(groupIndex)
            wordIndex += 1
    return groupIndices
```

2.) Ermitteln des Startindex der maßgeblichen Vokalgruppe in einem Wort

```
govIndex = 0
if len(groupIndices) == 0:
    endGroups.append(None)
else:
    if len(groupIndices) == 1:
        govIndex = groupIndices[0]
    else:
        govIndex = groupIndices[len(groupIndices)-2]
```

3.) Speicherung passender Endgruppen

```
if govIndex >= (len(word) * 0.5):
    endGroups.append(None)
else:
    endGroup = ""
    endIndex = govIndex
    while endIndex < len(word):
        endGroup += word[endIndex]
        endIndex += 1
    endGroups.append(endGroup)
```

4.) Speicherung der alle Kriterien erfüllenden Reimwörter

```
if endGroups[primEndIndex].upper() == endGroups[secEndIndex].upper():
    if (words[primEndIndex].upper() not in words[secEndIndex].upper() and
        words[secEndIndex].upper() not in words[primEndIndex].upper()):
        solutions.append(f'{words[primEndIndex]}, {words[secEndIndex]}')
```