

Aufgabe Störung:

Lösungsidee: Die Idee ist, dass man das erste Wort der Textstelle mit Störung als Schlüsselwort festlegt und dann nach diesem Schlüsselwort im Text sucht. Wenn dann das Wort gefunden wurde, wird überprüft, ob die darauf folgenden Wörter der Textstelle mit dem Text übereinstimmen. Falls sich an dieser Stelle jedoch die Störung bzw. ein Unterstrich befindet, wird zu dem nächsten Wort gesprungen, jeweils im Text sowie in der Textstelle. Wenn ja, wird die vollständige Textstelle auf der Konsole ausgegeben.

Umsetzung: Als erstes wird der Text ins Programm geladen. Da der Text nicht einheitlich formatiert ist, wird er erstmal einheitlich gemacht. Zusätzlich werden Sonderzeichen und Zeilenumbrüche entfernt.

Filterfunktion:

```
def filter_text(zeile):  
    # alle Zeilenumbrüche entfernen, die separat sind  
    for i in range(len(zeile)):  
        if zeile[i] == '\n':  
            zeile[i] = '  
  
'  
  
    # alle Zeilenumbrüche entfernen, die mit String verbunden sind  
    # Sonderzeichen entfernen  
    text = ""  
    verbotene_zeichen = ["?", "!", ", ", ">>", "«", "„", "“", "„", "[", "]", "(, ")  
    for phrase in zeile:  
        phrase.strip()  
        for zeichen in verbotene_zeichen:  
            phrase = phrase.replace(zeichen, "")  
        text += phrase  
  
    return text
```

(Zur Erläuterung nehme ich dieses Beispiel:)

```
text = "Hallo, ich bin Lars Wisotzky. Ich komme aus Deutschland."  
stoeuerung = "ich      "
```

Im nächsten Codeblock wird die Textstelle in ein Dictionary umgewandelt.

→ {0: 'ich', 1: '_', 2: '_', 3: '_'}

Anschließend wird nach dem Anfangswort der Textstelle im Text gesucht und der Index, wo das Anfangswort im Text gefunden wurde, in einer Liste gespeichert. Das würde dann so aussehen: "Hallo, ich bin Lars Wisotzky. Ich komme aus Deutschland." → [1, 4].

Im nächsten Schritt wird Störung ohne Unterstriche gespeichert. → ['ich']

Das ist wichtig, weil damit die mögliche, komplette Textstelle (Output des Algorithmus) verglichen wird.

Der Algorithmus:

```
# Wiederholt so oft wie es das Anfangswörter im Text gibt
for index in range(len(anfangswort_index)):
    # Ende der Wortgruppe bezogen auf dem Text
    ende_index = anfangswort_index[index] + len(txtstelle_dict)

    # möglicherweise komplette Wortgruppe (zum Vergleichen der Ergebnisse)
    algo_output = []
    # möglicherweise komplette Wortgruppe (für Ausgabe)
    komplette_wortgruppe = ""

    zaehler = 0
    # Jedes Wort was sich im Bereich des Anfangswortes (Index) und dem Ende Index befindet
    for item in text[anfangswort_index[index]:ende_index]:
        if item == txtstelle_dict[zaehler]:
            algo_output.append(item)
            zaehler += 1

        komplette_wortgruppe += (item + " ")
    # hier wird das Ergebnis des Algorithmusses mit der Störung ohne Unterstriche verglichen
    if algo_output == erwartete_woerter:
        print(komplette_wortgruppe)
```