Praktikum Programmieren - #2

Programmieren

1. Hackerize function

Erstelle mit Python eine Funktion `hackerize()` die einen übergebenen String in eine *hackerized* Version umwandelt.

Grundstruktur der Funktion:

```
def hackerize(text):
    """Transform text by changing several letters.
    Changes are: e => 3, i => !, o => 0, s => 5
    """
    pass
```

Anwendungsbeispiel:

```
print(hackerize("I write my text like passwords")) # --> I wr!t3 my t3xt l!k3
pa55w0rd5
```

Bonus:

Eine Möglichkeit diese Funktion zu erstellen und noch deutlich flexibler zu machen wäre, ein Dictionary mit den gewünschten Zeichenänderungen zu nutzen. So könnten Nutzer*innen die Funktion auch nach ihren Wünschen anpassen.

Ein Startpunkt sähe dann z.B: so aus:

```
def hackerize(text,
              replacements={"e": "3", "i": "!", "o": "0", "s": "5"}):
    """Transform text by changing several letters (as defined in replacements).
    """
    pass
```

2. Textanalysis

Bisher haben wir mit relativ kleinen Strings gearbeitet (da wir immer alles selber getippt haben). Jetzt werden wir mit einer deutlich größeren Datenmenge arbeiten und zwar einem ganzen Buch!

Hierfür nehmen wir das Buch "Der Mensch eine Maschine" von Julien Offray de la Mettrie. (Textdatei namens `de_la_mettrie_mensch_maschine_UTF8.txt` wird im Praktikum und/oder über Moodle bereitgestellt).

Dazu erstellen wir zuerst einige Funktionen die wir später benötigen:

read_text_file()

Erstelle eine Funktion die eine Textdatei einließt und als ein String zurückgibt.

```
def read_text_file(filename):
    """Import text file and return as string.
    """
    pass
```

text_to_words()

Für viele Analysen ist es hilfreich den Text in kleinere Teile aufzuteilen, z.B. in die einzelnen Wörter. Schreibe eine Funktion die einen String in einzelne Wörter aufteilt und eine Liste der Wörter zurückgibt. **Achtung:** Wörter sind nicht nicht einfach nur durch Leerzeichen getrennt sondern müssen auch noch auf einigen anderen Wegen bereinigt werden.

```
def text_to_words(text):
    """Converts a single text string into a list of words.
    """
    pass
```

Aufgabe 2.1

Erstellen sie ein Programm das mit Hilfe der Funktionen `read_text_file()` und `text_to_words()` die beiden folgenden Werte berechnet:

- Wie viele Wörter sind im Buch "Der Mensch eine Maschine" insgesamt enthalten?
- Wie viele *verschiedene* Wörter sind in diesem Buch enthalten?

Aufgabe 2.2

Jetzt soll der Text noch etwas genauer analysiert werden. Dazu soll eine Funktion erstellt werden, die zählt, wie oft jedes Wort im Text vorkommt. Ein dafür gut geeigneter Datentyp ist das *Dictionary*. Dazu könnten wir z.B. ein Dictionary `vocabulary` erstellen, beim dem jeweils das Wort als *key* und die Anzahl als *value* festhalten wird.

```
def word_counter(text):
    """Computes the 'vocabulary' of an input text string.
    vocabulary is a dictionary which contains all words in the text
    as well as the number of occurences of each word.
    """
    words = text_to_words(text)

    # add own code here

    return vocabulary
```

- Wie oft kommen die folgenden Wörter im Text vor: "Mensch", "Maschine", "das", "Das" ?

Aufgabe 2.3

- Was sind die 10 häufigsten Wörter im Text? Und wie häufig kommen sie jeweils vor?

Debugging

Beim Programmieren ist es völlig normal, dass das Schreiben von Code nur einen kleinen Teil der Zeit einnimmt. Oft geht mehr Zeit ins Land um den geschriebenen Code auch wirklich zum Laufen zu bringen. Damit wären wir beim Thema: Debugging!

Hier eine weiter Funktion auf die die Welt schon lange warten musste, die aber leider irgendwie so nicht funktioniert.

Bitte einmal "reparieren".

```
import random


def string_scatterer(text, space=" "):
return "".join([f"{c}{space}" for c in text])


def text_randomizer(list_of_words)
    final_string = ""

    for word in list_of_words:
        random_choice = random.randint(1, 4)
        if random_choice == 1:
            final_string = final_string + " " + word.upper()
        elif random_choice == 2:
            final_string = final_string + " " + word[::-1]
        else random_choice == 3:
            final_string = final_strig + " " + string_scatterer(word, "--")
        else:
            final_string == final_string + " " + string_scatterer(word.upper(),
"*")

    return final_string


word_list = ["Wow!", "This", "seems", "like", "a", "super", "useful", "tool!"]

for _in range(5):
    print(text_randomizer(word_list))
```