1] Create a list of strings (say, five), then use the function **sum** on it. You would probably expect the end result to be the concatenation of all strings, but, instead, the program crashes.
Show the result, and then <u>explain</u> what happened.

- Hint: Investigate the function **sum**.

2] Sometimes, we need to split a string starting from the end, i.e. from right to left. Search the official Python documentation[1] to find a method that does this. Use the method you found to create a list of only the <u>last</u> four words in this string:

```
st = 'It was the best of times, it was the worst of times'
['the', 'worst', 'of', 'times']
```

3] Use a **while** loop to return <u>all</u> occurrences of a substring in a string. The program should print the <u>index</u> and the substring for each occurrence. For testing, use:

- The string "Don't walk behind me; I may not lead. Don't walk in front of me; I may not follow. Just walk beside me and be my friend."[2]
- The substring "walk".

4] ► Use the string containing the text of T.S. Eliot's *The Waste Land*, created in class[3]. Find out how many times each of the substrings **the** and **is** occur in it (all possible capitalizations included!)

5] Based on the file **cities_distances.txt**, create a new file **cities_distances_4.txt**, in which every line has <u>four numbers</u> instead of two: the original two, their sum, and their product. For example, the first line is originally

```
7.682064339        715.3540281
```

and it has to become

```
7.682064339        715.3540281        723.0360924390001        5495.395669027014
```

Note: Remove the empty lines from the original file.

Dictionary problem (pls. use only a "plain" dictionary, not other data structures, like Counter!)

6] Given a string, create a dictionary whose keys are the characters of the string, and whose values are the nr. of occurrences of those characters in the string. Print the dictionary <u>in sorted order of the keys</u>. For example, if the string is 'bbcaaa', the program should output

```
a 3
b 2
c 1
```
.

---

[1] https://docs.python.org/2/library/string.html
[2] Albert Camus
[3] The text file can also be downloaded from our webpage.