בס"ד

Instructions for the submission of Homework Nr. 4 in Python

1) Similarly to what we already wrote in the instructions for the submission of the previous homework, you are requested to write the code that deals with keyboard input, in the following way:
   - In the case of getting the name (or pathname) of a file, you are requested to write the call to the *input* function, in the following way:

     ```
     file_name = input("Enter filename or fullfilename: \n")
     ```

   - etc.

   Note that the character "\n" at the end of the above string, that the *input* function prints, is very important. If that character will be missing at the end of the printed string, may be that the automatic checker will say that the specific run case that it tried to run, failed.

2) With respect to the printing of key-value pairs (the items) of a dictionary: in order to be sure that the printed output produced by your program will be in agreement with what the automatic checker expects, apply the *sorted* built-in function on the *items()* of the dictionary. For example:
   - Let us assume that the options of the menu of the program in question Nr 3 is are stored in a dictionary like the following:

     ```
     func_names = {'1':"sortedzip", '2':"reversedzip"}
     ```

   - In order to be sure that those menu options will be correctly printed, you will need to sorted them by the key of the dictionary. To do so, apply the built-in *sorted* function on the *items()* of the dictionary, in the following way:

     ```
     sorted(fnames.items())
     ```

3) With respect to question Nr 4: In this question, the program needs to generate sentences for which the words that compose them must be randomly retrieved from sequences of syntactically corresponding words. That means that this question is not adequate for the automatic checker, that expects non-random outputs. Because of that, a regular submission box was created for it.

4) Here you have run examples for all the questions in this homework. Those examples show you how the automatic checkers expect the input and output of the programs, to be.

- Question Nr 1
  - Run example (your program needs to import the provided module *Targil4input.py*):

    Aharoni, [12345], 75, 0
    Kaner, [23415, 12345], 68, 9.899494936611665
    Korman, [], 0.0, 0.0
    Melamed, [23452, 23459], 86.5, 4.949747468305833
    Zloti, [23560, 23452], 90.5, 13.435028842544403
    80.71428571428571, 12.365466505747444

- Question Nr 2
  - Run example:

    Let us assume that the input text file contains the following lines of text:

    Now, fair Hippolyta, our nuptial hour.
    Draws on apace: 4 happy days bring in
    another moon; but Oh! me thinks how slow
    this old moon wanes; she lingers my 3 desires,
    Like to a step dame, or a dowager
    long withering out a young man's 2 revenues.

    The printing of the run's result will be as follows:

```
Enter filename or fullfilename:
sh1.txt
LineNr    nr of vowels    nr of b-m consonants    nr of n-z consonants
  0           13                5                       12
  2           11                9                       11
  4           11                6                       7
Nr of Lines of text    total nr of vowels    total nr of b-m consonants    total nr of n-z consonants
3                     35                      20                           30
```

- Question Nr 3
  - Run example A:

    Enter the size of the sublists of the list to process:
    4
    Enter the list to process:
    [[2,1,6,'aa'],['bcd',5,8,7],[10,21,30,3]]
    1: sortedzip
    2: reversedzip
    Which function do you want to choose?
    1
    [(1, 5, 10), (2, 7, 21), (6, 8, 3), ('aa', 'bcd', 30)]
    [[1, 2, 6, 'aa'], [5, 7, 8, 'bcd'], [10, 21, 3, 30]]

  - Run example B:

    Enter the size of the sublists of the list to process:
    4
    Enter the list to process:
    [[2,1,6,'aa'],['bcd',5,8,7],[10,21,30,3]]
    1: sortedzip
    2: reversedzip

```
Which function do you want to choose?
2
[('aa', 7, 3), (6, 8, 30), (1, 5, 21), (2, 'bcd', 10)]
[['aa', 6, 1, 2], [7, 8, 5, 'bcd'], [3, 30, 21, 10]]
```

Note that the last line, in both run examples, is the result of applying the function unzippy on the result returned by sortedzip or by reversedzip, correspondingly.

- Run example C:

```
Enter the size of the sublists of the list to process:
4
Enter the list to process:
[[2,1,6,'aa'],['bcd',5,7],[10,21,30,3]]
ERROR - all sublists must be of size 4
```

- Run example D:

```
Enter the size of the sublists of the list to process:
4
Enter the list to process:
[[2,1,6,'aa'],['bcd',5,8,7],[10,21,30,3]]
1: sortedzip
2: reversedzip
Which function do you want to choose?
3
ERROR - chosen function does not exist.
```

- Question Nr 4

Let us assume that the tuples containing the words that will be used by your program to create each line of the random poem are as follows:

```
peopleNames = ("Iosi", "Ety", "Jack")
verbs = ("sees", "plays", "sings")
adjectives = ("tall", "small", "red")
adverbs = ("slowly", "tomorrow", "now", "soon", "suddenly")
animateObjects = ("flowers", "oranges")
inanimateObjects = ("stones", "chairs")
```

- Run example A:

```
How many lines do you wish the poem to have?
5

Iosi now plays tall stones
Iosi tomorrow sees small chairs
Jack now sees small chairs
Iosi slowly sees small flowers
Ety soon plays small stones
```

- Run example B:

```
How many lines do you wish the poem to have ?
0
ERROR: it must be a positive integer
```