COSC 1336
Spring 2023

Assignment 8: Strings and Text Processing

[1] **Objectives**: This assignment aims to practice using strings and their many methods. Text processing is a prevalent task in Computer Science. Please read about the methods that we are using in this assignment. An additional purpose of this assignment is to practice dividing a project into smaller modules.

[2] **Description**: The assignment is to develop a program that processes a text and extracts all words out of the text into a word list. In the process, delete spaces, end-of-line characters, and punctuations. From the word list, produce a list of word-count pairs. The pair is a list with a word as the 0-th element and a count as the 1-st element. The count represents the number of occurrences of the word in the text. A detailed description is given below.

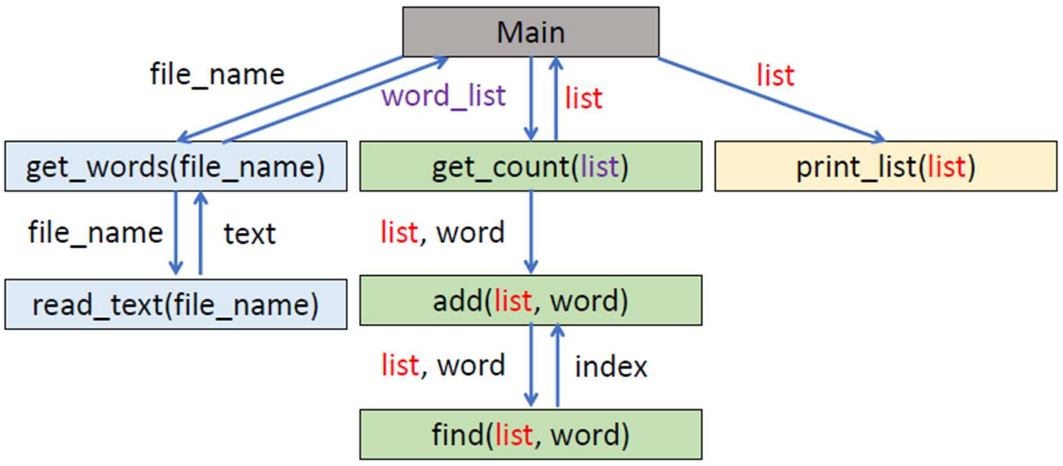
- Since we are not ready to deal with files, the code is below. You will be given a function to extract all characters from a file and store them in a string. You must call the function with a physical file name and return a string with all characters. The code is simple enough to understand.
- You should have a main program that
    - Ask the user to input a file name.
    - Get all words from the file into a list of words. A word here is a non-empty string that does not begin or end with a punctuation sign and cannot be all digits. For example, “`don't`”, “`A1`”, “`4ever`” are words, but “`**`” and “`2023`” are not. We have to use this definition to keep the code simple.
    - There is a `string.punctuation` that is a string of all punctuations. Remember to import the string module.
    - Create a list of lists (wc_list for word-count list) showing the number of occurrences of all the words in the list. For example, [[“apple”, 2], [“banana”, 1]]. For this purpose, we will ignore the cases of a letter by converting all words into lower cases. Here is the strategy: for each word in the word list, add it to the wc_list if the word has never been added before. The count for this word will be set to 1. Otherwise, increase the count by 1.
    - Print the wc_word list in alphabetical order.
    - Print the wc_word list in decreasing order of the count.

```
def read_text(file_name):
    f = open(file_name)
    text = f.read()
    f.close()
    return(text)
```

[3] **Requirements:** You must write the program described in this assignment, including all the following functions.

- `find(list, word)`: given a two-column table, find the word in the 0-th column. Return the index of the word if found. Otherwise, return None.
- `add(list, word)`: add the `word` into the 2-column list with the word followed by the count, or increase the count by one as specified above. You must call the find function above. The list will be updated in place one way or another after the call. There is nothing to return.
- `print_list(list)`: prints the 2-column table with count first. (See sample output). There is nothing to return.
- `get_words(file_name)`: extracts all words out of the string. You will use several string methods to do the job in this function. The methods are there for you to use. Don't write low-level code. The function returns a list of words.
- `get_count(list)`: takes the 1-column list and produces the two-column list. You must use the `add` function to add a word to the list. The function returns the 2-column list.

The following figure shows the structure of the program. We can make each function simple and short by breaking it up into smaller pieces. The longest function is six lines in my program. It may take you a few extra lines each.



This program can be done in about 50 lines using Python's string methods and features, such as list comprehension.

There are a few tricky issues, and I will point them out in class.

[4] **Output**: A sample output is given below. The list is long, so I showed only the first 50 lines. You should test for all possible input. You should note that I added [ ] around the words to ensure I am not getting some white characters in the word.

[5] **Deadline**: 11:59 pm, Monday, April 7, 2023

```
Enter the text file name to process: GettysburgAddress.txt
```

```
Count | Word
-----   ---------------
    7 | [a]
    1 | [above]
    1 | [abraham]
    1 | [add]
    1 | [advanced]
    1 | [ago]
    1 | [all]
    1 | [altogether]
    6 | [and]
    1 | [any]
    3 | [are]
    1 | [as]
    1 | [battle-field]
    2 | [be]
    1 | [before]
    1 | [birth]
    1 | [brave]
    1 | [brought]
    2 | [but]
    1 | [by]
    5 | [can]
    1 | [cause]
    1 | [civil]
    1 | [come]
    2 | [conceived]
    1 | [consecrate]
    1 | [consecrated]
    1 | [continent]
    1 | [created]
    3 | [dead]
    2 | [dedicate]
    4 | [dedicated]
    1 | [detract]
    2 | [devotion]
    1 | [did]
    1 | [died]
    1 | [do]
    1 | [earth]
    1 | [endure]
    1 | [engaged]
    1 | [equal]
    2 | [far]
    1 | [fathers]
    1 | [field]
    1 | [final]
    1 | [fitting]
    5 | [for]
    1 | [forget]
    1 | [forth]
    1 | [fought]
    1 | [four]
    1 | [freedom]
    2 | [from]
    1 | [full]
    2 | [gave]
    1 | [god]
    1 | [government]
    3 | [great]
    1 | [ground]
```

```
Count | Word
-----   ---------------
   13 | [that]
   11 | [the]
   10 | [we]
    8 | [here]
    8 | [to]
    7 | [a]
    6 | [and]
    5 | [can]
    5 | [for]
    5 | [have]
    5 | [it]
    5 | [nation]
    5 | [not]
    5 | [of]
    4 | [dedicated]
    4 | [in]
    4 | [this]
    3 | [are]
    3 | [dead]
    3 | [great]
    3 | [is]
    3 | [people]
    3 | [shall]
    3 | [so]
    3 | [they]
    3 | [us]
    3 | [who]
    2 | [be]
    2 | [but]
    2 | [conceived]
    2 | [dedicate]
    2 | [devotion]
    2 | [far]
    2 | [from]
    2 | [gave]
    2 | [living]
    2 | [long]
    2 | [men]
    2 | [new]
    2 | [on]
    2 | [or]
    2 | [our]
    2 | [rather]
    2 | [these]
    2 | [war]
    2 | [what]
    2 | [which]
    1 | [above]
    1 | [abraham]
    1 | [add]
    1 | [advanced]
    1 | [ago]
    1 | [all]
    1 | [altogether]
    1 | [any]
    1 | [as]
    1 | [battle-field]
    1 | [before]
    1 | [birth]
```