CSCI203 ASSIGNMENT ONE
Due 23:55 Friday 20/08/2021

Task:

You must write a program which reads, processes and reports on the contents of a text file.

Your program should:
1. Read the name of the text file from the console.
2. Read in a text file, not all at once. (This can be line by line, word by word or character by character.)
3. The file content must be converted to a sequence of words, discarding punctuation and folding all letters into lower case.
4. Store the unique words and maintain a count of each different word.
5. The words should be ordered by decreasing count and, if there are multiple words with the same count, alphabetically. (This ordering may be achieved as the words are read in, partially as the words are read or at the end of all input processing.)
6. Output the total number of words and the number of "unique words"
7. Output the first fifteen words in the sorted list, along with their counts.
8. Output the last fifteen words in the list, along with their counts.

Implementation Requirement:

You must choose **appropriate** data structures and algorithms to accomplish this task. *Note that*
1) **in the context of this assignment, appropriate choices will be efficient and will not use excessive instructions or data**.
2) where a punctuation mark appears between two letters, the sequence is to be treated as a single word. Thus, *it's* will become *its*, *you'll* will become *youll* and *loop-hole* will become *loophole*.
3) you can assume that the input file contains no more than 50,000 different words.
4) Two sample input files "`sample-short.txt`" and "`sample-long.txt`" is provided for you to test your program and produce output.
5) you may use any data structures or algorithms that have been presented in class up to the end of week 4. If you use other data structures or algorithms appropriate references must be provided.
6) Programs must compile and run under `gcc` (C programs), `g++` (C++ programs), `java` or `python`. Programs which do not compile and run will receive no marks.
7) Programs should be appropriately documented with comments.
8) All coding must be your own work.
9) Standard libraries of data structures and algorithms such as STL **should not be used**.
10) Code be sourced from textbooks, the internet, etc may also not be unless it is correctly credited. In the event that you use code sourced in this way you will not receive marks for that part of the program.

Report:

A pdf file describing your solution and program output should be produced. This file should contain:
1. A high-level description of the overall solution strategy.
2. A list of all of the data structures used, where they are used and the reasons for their choice.
3. A list of any standard algorithms used, where they are used and why they are used.
4. The output produced by your program on the provided "`sample-long.txt`" file.
5. The report should be no more than 2 pages. If it is more than 2 pages, marking will be only based on the first two pages.
6. The report pdf file should be called `<your login>-a1.pdf`

Marking Guide:
- Programs submitted must work! A program which fails to compile or run will receive a mark of zero.
- If your program produces different output from what is reported in the pdf file, a mark of zero will be graded.
- A program which produces the correct output, no matter how inefficient the code, will receive a minimum of 50% of the program component of the mark.
- Additional marks beyond this will be awarded for the appropriateness, i.e. efficiency for this problem, of the algorithms and data structures you use.
- Programs which lack clarity, both in code and comments, will lose marks. The total mark will be determined based on both your code and the accompanying design pdf document.

Submission via Moodle:
- Source code for the assignment should be typed into a single file called `<your login>-a1.ext` where *ext* is the appropriate file extension for the chosen language.
- Your submission will consist of two files:

  `<your login>-a1.ext` and `<your login>-a1.pdf`

  where *ext* is one of `c`, `cpp`, `java` or `py`. `<your login>` is your UOW login name