Work with ONE other team member to complete this assignment -- (same team member as Milestone #1 & 2)!  Only ONE person has to upload it to this Blackboard submission hyperlink for grading -- type both person's names into the submission comments.

End goal -- create a Python program that takes as file input the patent application text file your team created in Milestone #2, manually perform Bag of Words text processing on the input to extract keywords (important concepts) from that unstructured text input, and output the index. html mockup you created in Milestone #1.  Read under this Blackboard section the instructions to perform Bag of Words text processing -- do not use the nltk external library to do the work for you (that comes later).

You’ve received feedback from your internal client, and she has added another requirement based on the work you’ve done so far on this project.   The reporting functionality you prototyped in Milestone #2 is good, but let's set it aside for now.

She likes your mockup of an index. html file produced by Milestone #1 (which lists keywords, or important concepts, extracted from the patent application data), but now she realizes that the keywords should be sorted in ascending order on the index. html page.  Additionally, your team created a feasible patent application inventory input file to work with in Milestone #2, so your team should not hard code the keywords into a Python program that produces index. html, but instead use file I/O to read the patent application text into a Python program from the input file, perform text analysis using the delete noise (junk words) and substitution (word replacements to standardize your text) lists she provides (lists which you may augment and change as your team sees fit) to pull out keywords (concepts) from the input file, and produce the index. html file that your team started to create in Milestone #1.  Since you are developing this in a RAD environment, you should not create a production-ready program — this should be a partial solution that you continually augment as you move through the milestones — so keywords have to be “dead” hyperlinks and you may augment your delete and substitution lists when submitting future milestones.

Be sure to include all partners’ names in the submission comments on Blackboard and in your so urce code comment block.  Submit your work as a . py file: “ Milestone3A\_ Full name1\_ Full name2. py” ( fullnames are placeholders for your each person's full name ).  Also submit your input file, deletion and substitution lists:  "Milestone3 AInput\_Full name1\_ Fullname2. txt" and " Milestone3ASub\_Full name1\_Full name2. txt" and " Milestone3ADel\_Full name1\_Full name2. txt."  For the A-level grade, also be sure to submit "Milestone2\_Fullname1\_Fullname3A.docx"  (fullnames are placeholders for your each person's full name). NO late work will be accepted.  **Due: ~~Friday, 2/13/17~~ Wednesday, 4/19/17 @ 11:59pm**

C-Level: (70%) Organize the program that you created and the data that you have collected.  You will be graded on the successful implementation.

B-Level: (90%) Be sure to document your code and data well.  You will also be graded on how well you can describe how it works.  The comments and documentation required here exceed the typical documentation that you would provide in your code to allow another programmer to modify it.  You need to show that you understand how it is working and why you make the choices that you did.  Always be sure, at the top of any source code that you submit, you include four pieces of information: authors' names, creation date, last modification date, and brief description of your source code.

A-Level: (100%) You should include a document that describes your understanding of how well the solution works and in what situations you can apply this to other problems.

**HINT: INVESTIGATE OPENING A FILE WITH APPROPRIATE ENCODING.  DEFAULT ENCODING DIFFERS ON DIFFERENT OPERATING SYSTEMS.  ALSO BE SURE TO RUN YOUR PROGRAM AT COMMAND PROMPT OR TERMINAL PRIOR TO SUBMISSION FOR GRADING.  IF YOU HAVE ENCODING ISSUES, POST TO THE TECHNICAL DISCUSSION BOARD FOR HELP OR VISIT A TA DURING OFFICE HOURS.**

Please follow the guidelines mentioned in the below link:   
<http://stackoverflow.com/questions/24616678/unicodedecodeerror-in-python-when-reading-a-file-how-to-ignore-the-error-and-ju>

In brief, ensure errors are set to "ignore" as per the code snipped below:

 file = open(filename ,'r', encoding="utf-8", errors="ignore")