* The purpose of this lab is to store text messages in an efficient method.
  + The goal is to not store the same word twice, from all of the messages.
  + Consider using arrays, arraylists, or lists.
* Student has a choice of languages: C, C++, Python, or Java.
* This program can be a Windows or Console app. A description of the window will be supplied.
* In your program, create a comment header, at the top of the document, stating information about this lab. Include:
  + Your name
  + CmpSc472, Final Lab
  + One- line description of the lab

Program Requirements

1. What ever language used, name the project/class “**CmpSc472\_MsgMapping”.**
2. Input: The messages will come from a text file. Create your own text file with messages.
   1. I will use my own text file to grade your lab.
   2. This text file will contain at a minimum of 20 messages.
   3. Read each message, one line at a time.
   4. Messages shall contain status information about a specific topic.
      1. **The alarm is set.**
      2. **The alarm is disabled.**
      3. **Emergency switch is disabled.**
3. Processing
   1. Part1: Reading in the messages.
      1. Read messages from the text file. Ask the user to input the name of the text file.
      2. Parse the string and store **each word** into an array, list, array, etc.
      3. You will probably need supporting arrays/lists that keep track of (1) The words position in the array, (2) The # of words in a message, and (3) The position of the words for each message.
      4. This part is finished when all messages have all of the words stored into the arrays/lists, with supporting info in other arrays/lists.
   2. Part 2: Retrieving a message.
      1. After messages are read in, the user will ask to enter the message number they wish to retrieve.
      2. Validate the number. It must be between 1 and the max message stored.
      3. If valid, retrieve all of the words that comprise that message.
      4. Put the message back together and display the message.
      5. Loop and ask for different messages until user is finished.
4. Windows App Option
   1. After Part 1, the Windows app will display
      1. The file used to read in the text messages.
      2. Show a count of the number of messages read in.
      3. Show a count, and all of the different words read in.
   2. Part 2,
      1. The message user wants to display.
      2. The re-assembled message requested.

Deliverables

* Submit the appropriate file, or compressed folder, based on the programming language and IDE you used.
* You can submit your message text file, as well, but I have my own to test with.

If you decide to makes a windows app for this lab, this window might be a good design for your window.

