You will enhance your GenericArray code from Lab14. I have provided my solution for GenericArray you will add 4 methods in GenericArray.c. You will also need to complete the methods for word.c. Please read the documentation for each in the appropriate .h file.

Specific Information
- You will not use gets – use scanf, fscanf, fgets
- For the specific functions I have provided documented .h files. The comments are Javadoc style. Look through the .h files. You CANNOT change the .h files in any fashion!
- I have provided the menu functions and the fileUtils/utils functions.
- I have provided a Makefile you must use.
- You will create an output file that shows the run of the program using valgrind. This output file should show that no memory is being leaked.
- I have provided unchangeable cscd240Lab15.c

What You Must Complete
- You must complete the functions in word.c (See word.h)

    A word contains
    - char * ltrs
    - int len – number of characters in the word

- You must complete the functions in genericArray.c (See genericArray.h)
    - addItem
    - removeItemByValue
    - removeItemByIndex
    - removeItemByIndexPassedIn

FILE SPECIFICATIONS FOR THE FILE
word
word
...
word

TO TURN IN
1. A zip file, in Canvas, by the required due date containing your Lab15 folder and all its code including
    - All yours and my code including my Makefile
    - All yours and my input file(s) used to test your program
    - All yours and my output file(s)
    - A full test run named cscd240Lab15val.txt – testing all aspects including showing your code is leak free.

We should be able to download the zip, unzip it, type make and compile your code, then run your code.

Name your zip, your last name first letter of your first name lab15.zip (Example: steinerslab15.zip)