**Project #1**

Write a program that can maintain a roster of *Student*, using a *LinkdStudentList* to enter students originally, and then storing the students into a sorted array. with a maximum of 10 students in the roster. The program should be able to search for a given student, add a student, and drop a student. The roster should be saved as a text file. The data type should be *Student* for this part. (We will use the polymorphism for *StudentRecord* in Project #2*.)*

Write your ow *LinkedStudentList* class, including *insertSorted(Student s), delete(Student s), isEmpty, toString()* You will create a sorted array for the roster from the list when the user selects “Save Changes”

Your array should have a max size of 10 for this part of the project, but should be partially filled to begin. After the students have been stored in the sorted array, you should allow the user to add students (maintaining sorted order, but not allowing more than 10 students), drop students (with no “null” array entries) and search for students by ID and by name. Your search should return the *Student*, not a Boolean.

You should include a menu for the user, including:

Load roster from a file   
Add student  
Remove student  
Search student by ID (*.equals*)   
Search student by name (*.compareTo*)  
“Save” change (this method will copy the list to a sorted array)  
“Save changes” (this method will save a sorted text file in a different file, so I can see the final roster)

You may use a text file to store the original students (to make it easier for you to test) and load that unsorted roster into the linked list.

You should be able to add/remove students using both the list and the array, **but not both at the same time**. Once the array is created, you will not need the list! (Your final part of the semester project will include “documentation’ discussing the differences in using the list versus the array for adding/removing.)

Please submit the project either by attaching the separate files, or “zipping”, but please do not send me “jar” – I have had some trouble opening those (I think it is my Kaspersky causing the problem, but I haven’t resolved it yet.) Never send the .class files!

This part of the Semester Project will be due on Monday July 26, with Project #2 (which will add a waiting list) due Friday July 30.