**The School Database**

Acme University has contracted you to develop a rudimentary database program to track their students. This program must adhere to the following specifications:

1. The program, when first started, must prompt the user for the name of the database to use.
2. The program must check to see if the file already exists on the computer system. If the file does not exist, the program must create the file and notify the user that it has done so. If the file does exist, the file is used for data input.
3. All data from the file mentioned above must be read and placed into a vector in alphabetical order by the student's last name. Data, all of which is in "string" format and on a single line in the file for each stidemt, should consist of the following fields (which are to be separated by colons within the file and vector, but not on the report):
   * Student's last name.
   * Student's first name
   * Student's major
   * Student's current G.P.A.
   * Student's advisor's name
   * Student's advisor's phone number
   * Student's transfer objective
4. Once the data is into the vector, the user of the program should be given the following choices to manipulate the data:
   * Insert a new person into the database. Insertion of new data must place the new data in alphabetical order by student's last name.
   * Delete a student.
   * Edit student's data
   * Search for a student by last name.
   * Display all students in the vector
   * "Print" a formated roster with column headers, student's last name in all capitals,, a comma, their first name with only the first letter capitalized, a space, a colon, then the data (in aligned columns). Advisors name must be in format of having the first letter of the first name and the last name capitalized, all other characters in lower case.
   * Exit. Upon exiting, the original data file shall be replaced with the contents of the vector, with each line of the data file holding the above information for each student. The data file must be in text-only format.

Extra credit will be given for the following:

1. Proper handling of students that have the same last name, but different first names. (That is, proper insertion into the list in alphabetical order, proper deletion of students, etc.). This is worth 50 extra credit points.
2. Allow the user, when "printing" to have several selections: (This is worth 50 points)
   * Print sorted by GPA, then by last name of the student
   * Print sorted by advisor. then by last name of the student
   * Print sorted by major, then by last name of the student