Part A of this first programming project consists of providing the analysis and design for the following problem statement. **The work must be entirely your own**, except for any help you may get from the instructor. Hand in a clean, readable copy of your analysis and design. The implementation portion of this project assignment will be handed out March 21/22 and will be due on April 2/3. There will be additional implementation criteria not required of this analysis and design.

**Problem Statement**

Your instructor would like a program that will help him/her assign letter grades. The program will begin by asking the instructor for the number of students in his/her class. For each student, the program will read in a name and a list of scores that end with a negative number (that is not considered one of the scores). The program will then compute the average score for each student. Finally, the program will display the names and average scores sorted by the average score in descending order (highest to lowest), and assign a letter grade to each student based on a straight 90-80-70-60 scale. (I.e., average scores of 90 or better earn a letter grade of 'A', average scores of 80 or better and less than 90 earn a letter grade of 'B', and so forth.)

Write the (entire) analysis and design for this program that meets the design criteria below. Use the format shown in class and in the on-line handout *An Analysis and Design Style Guideline*.

**Design Criteria**

This program must be designed meeting the following criteria:

1. There must be two arrays, one for the names and one for the average scores.
2. There must be a function that reads in the data for each student (i.e., a name and the scores) and stores the name and the computed average score into the arrays.
3. There must be a function that sorts the two arrays so that they are in descending order by average score. This function must use Swap functions (as shown in lecture and an in-class exercise) to exchange elements of the arrays.
4. There must be a function that displays the list of students' names, their average scores, and their letter grades.
Of course, you may design other functions that are used by the required functions. For the purposes of this analysis and design, we will assume that a name consists of a single word without spaces. You may assume that there will never be more than 50 students and that the instructor will not try to enter data for more than 50 students. Note that the number of scores may be different for each student.

**Example Program Run**
The output might look something like:

**Welcome to CS 210 automated grade book program.**

**How many students are in your class? 5**

Please enter a student's name: Doe  
Please enter the student's scores (ending with a negative number)  
100 98 97 -1

Please enter a student's name: Dopey  
Please enter the student's scores (ending with a negative number)  
0 24 88 66 -1

Please enter a student's name: Hill  
Please enter the student's scores (ending with a negative number)  
44 99 100 100 -1

Please enter a student's name: Jones  
Please enter the student's scores (ending with a negative number)  
68 80 75 78 -1

Please enter a student's name: Smith  
Please enter the student's scores (ending with a negative number)  
69 85 76 -1

**The grades assigned are:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Average</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doe</td>
<td>98.3</td>
<td>A</td>
</tr>
<tr>
<td>Hill</td>
<td>85.8</td>
<td>B</td>
</tr>
<tr>
<td>Smith</td>
<td>76.7</td>
<td>C</td>
</tr>
<tr>
<td>Jones</td>
<td>75.3</td>
<td>C</td>
</tr>
<tr>
<td>Dopey</td>
<td>44.5</td>
<td>F</td>
</tr>
</tbody>
</table>