Programming Assignment #6B: Using Files, Arrays, and Structures

Adding the information of names into Programming 6A:

Read Chapter 11 of the textbook about ***structures*** as data types. We want to extend programming assignment #6A further to record not just the score of each student, but also the first name and the last name of each student. To do so, you have to include additionally the string library and define a new student record ***structure*** type *stdRecord* before your main function as shown below in red. Note that the first two data members in the *stdRecord* structure are string objects for storing the first name and the last name of a student, while the third data member in the *stdRecord* structure keeps the actual score.

```
…
#include <string>

struct stdRecord
{    string firstName;
     string lastName;
     double score;
};
…
```

Data Structures (variables and the array of structures) to retain the information in the main memory of the computer:

(i) Instead of using an array of double to store the scores in programming 6A, now you should use an array of *stdRecord* structures in your main function to record both names and scores of the students:

```
stdRecord myStdRecords[100];
```

(ii) You should still declare in your main function a variable *int numStudents* to store the information of the number of students. This allows the user to handle any number of students as long as it is no more than 100 students. In the beginning, let this variable be initialized to 0.

File format to retain the information in files: A file for holding the information of a test for Programming #6B should consist of numbers and names with each number and name on a separate line. The first number in the file should be the number of students in the class, followed by the first name, the last name, and the test score of the first student, and then those for the second student, and so forth, one by one. For e**xample**, the following would be the contents in a file **showing the names and scores of two students** in a class:

```
2
John
Brown
95
Maria
Cook
88
```

Requirements:

Revise your program for #6A accordingly to incorporate the information processing of both names and scores throughout the options provided by the program. **In addition, (i) for option *D* we would like to display the names of all the students with the highest and all the students with the lowest score respectively, and (ii) for option *Q* we would like to check with the user to see whether we should save the contents in** *numStudents* and *myStdRecords* into a file before closing the program and quit. See the details below.

- ***K***: the **K**eyboard option for the user to (i) manually enter the information of the number of students in the class and store it in *numStudents* and then (ii) for each student manually enter the *first name*, the *last name*, and the score of the student and store them in the data members of the corresponding *stdRecord* structure within the array *myStdRecords* accordingly.
- ***R***: to **R**ead from a file (with the file name entered by the user) the information of (i) the number of students in the class and store it in *numStudents* and then (ii) for each student read the *first name*, the *last name*, and the score from the file and store them in the data members of the corresponding *stdRecord* structure within the array *myStdRecords* accordingly.
- ***D***: to ***D***isplay **(i)** the number of student in the class and then for each student the name and the corresponding score of the student currently stored in the array, **(ii)** calculate and display the average of the scores, **(iii)** **calculate and**

display the highest score and display the names of all the students with the highest score (*note that there may be multiple students with that highest score*), and **(iv) calculate and display the lowest score and display the names of all the students with the lowest score (*note that there may be multiple students with that lowest score*),**

- ***M***: to ***M***odify one of the scores by re-entering a new score, a new *first name*, and a new *last name* from the keyboard,
- ***W***: to ***W***rite the number of students into a file (ask the user to enter the file name and read it from the user), and then for each student write the *first name*, the *last name*, and the score into the file, properly separating these prices of information into separate lines (or separating them with spaces in between).
- **Q**: to ***Q***uit the program. **However, before quitting program, your program should ask the user whether the user wants to save the contents in *numStudents* and *myStdRecords* into some file before closing the program. If the user wants to do so, the program should go through all the things required in option *W* to save the information into a file before closing the program.**