Lab Session 9: File Handling - Solution

Exercise 1: Writing Data as String and Character-by-Character to a File

Write a C program that asks user to input number of students, then ask the user to enter name of each student, and then write this data to file **studentnames.txt** by:
1) writing data as string.

2) writing data character-by-character.

Ensure student name appears in new line.

Exercise 2: Reading Data as String and Character-by-Character from a File

Using the rewind function, modify exercise 1 to read and display the contents of **studentnames.txt** 2 times, such that there is a space between each character of the student name in the 2nd scan.

Exercise 3: Writing Integer to a File

Write a C program that asks the user to input number of students, then ask the user to enter student score, and then write this data to file **studentscore.txt**.

Exercise 4: Reading and Append Integer

Write a C program that opens the file created in exercise 3, calculate the average score of all students, and append the average to file **studentscore.txt**.

Exercise 5: Creating leaderboard and storing to file

Given a file (**playerscore.txt**) that contains player names and scores as shown below, sort the score to show highest scoring player and their score at the top. Write the sorted score to file (**playerscore_sort.txt**).

playerscore.txt (you can manually create this text file with the following sample content.

Adam 200

James 70

Robert 110

Elena 230

Patrick 190

Brian 40

Matt 150

Exercise 6: Reading leaderboard and displaying on screen using ncurses

Write a C program to read sorted player names and scores from file **playerscore_sort.txt**. Using Ncurses, display the player names and score on the screen.