

**Computer Science Department**

**COMP133 ( Spring 2022)**

**Final Project *Due Date: Monday 6/06/2022 by 10:00 pm ( on Ritaj )***

In this project, you need to implement the major parts of the functions you created in phase one as follows:

***void displayMainMenu();*** // displays the main menu shown above

This function will remain similar to that in phase one with one minor addition which is the option:

***4- Print Student List***

***void addStudent( int ids[], double avgs[], int \*size);***

This function will receive the arrays containing the id numbers and the avgs as parameters. It will also receive a pointer to an integer which references the current size of the list (number of students in the list).

The function will check to see if the list is not full. If list is not full ( size < MAXSIZE) then it will ask the user to enter the student id (*four digit number you do NOT have to check just assume it is always four digits*) and then search for the appropriate position *( id numbers should be added in ascending order* ) of the given id number and if the id number is already in the list it will display an error message. If not, the function will shift all the ids starting from the position of the new id to the right of the array and then insert the new id into that position. Same will be done to add the avg of the student to the avgs array.

***void removeStudent(int ids[], double avgs[], int \*size);***

This function will receive the arrays containing the id numbers and the avgs as parameters. It will also receive a pointer to an integer which references the current size of the list (number of students in the list).

The function will check if the list is not empty. If it is not empty (size > 0) then it will search for the id number to be removed and if not found will display an error message. If the id number exists, the function will remove it and shift all the elements that follow it to the left of the array. Same will be done to remove the avg of the student from the avgs array.

***void searchForStudent(int ids[], double avgs[], int size);***

This function will receive the arrays containing the id numbers and the avgs as parameters. It will also receive an integer which has the value of the current size of the list (number of students in the list).

The function will check if the list is not empty. If it is not empty (size > 0) then it will ask the user to enter an id number and will search for that id number. If the id number is not found, it will display an error message.

If the id number is found then it will be displayed along with the avg in a suitable format on the screen.

***void uploadDataFile ( int ids[], int avgs[], int \*size );***

This function will receive the arrays containing the id numbers and the avgs as parameters. It will also receive a pointer to an integer which references the current size of the list (number of students in the list).

The function will open a file called ***students.txt*** for reading and will read all the student id numbers and avgs and store them in the arrays.

***void updateDataFile(int ids[], double avgs[], int size);***

This function will receive the arrays containing the id numbers and the avgs as parameters. It will also receive an integer which has the value of the current size of the list (number of students in the list).

The function will open the file called ***students.txt*** for writing and will write all the student id numbers and avgs in the arrays to that file.

***void printStudents (int ids[], double avgs[], int size); // NEW FUNCTION***

This function will receive the arrays containing the id numbers and the avgs as parameters. It will also receive an integer which has the value of the current size of the list (number of students in the list).

This function will print the information (ids and avgs) currently stored in the arrays.

***Note: You need to define a constant called MAXSIZE ( max number of students that may be stored in the ids and avgs arrays) equal to 100.***

***IMPORTANT NOTE: Your functions should have exactly the same number of parameters and types as described above and should use parallel arrays and work as described in each function. You are not allowed to use structures to do this project.***

**Items that should be turned in by each student:**

1. ***A copy of your main.c file***
2. ***An MSWord document containing sequential images of a complete run similar to the output shown on pages 4-8***

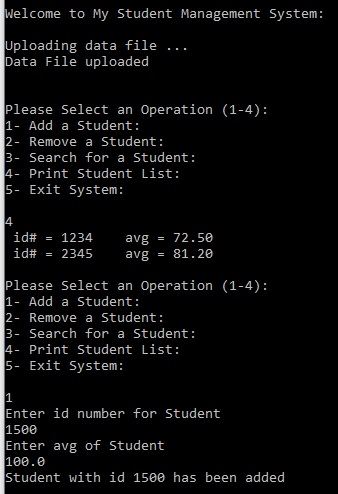
***SAMPLE RUN:***

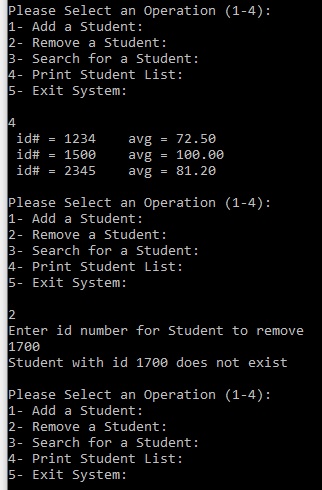
Make sure your program works **very similar** to the following sample run:

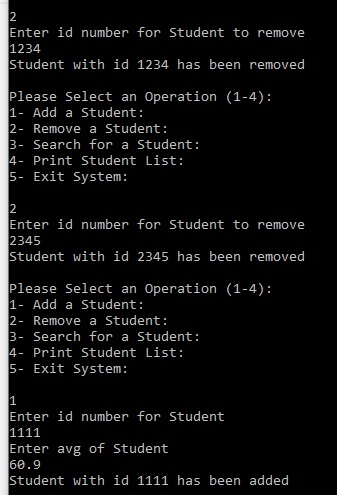
***Assuming that at the beginning of the run file students.txt has the following information stored (first column = ids and second column = avgs):***

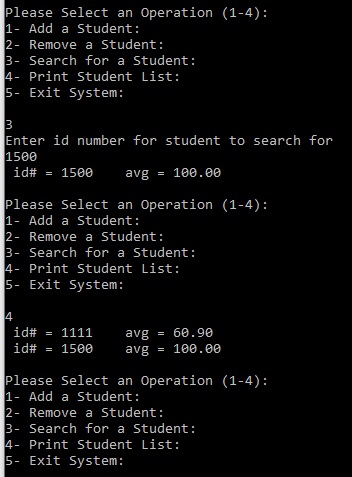
1234 72.5

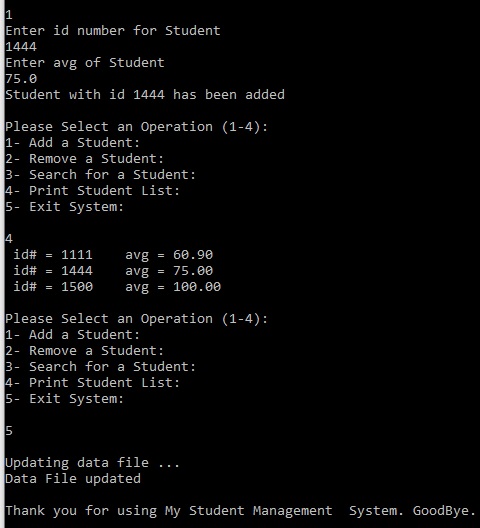
2345 81.2

******

******

******

******

******

***Late projects will not be accepted for any reason***.