Grading Guidelines for Assignments 17 and 18 (February 25)

<u>Deduct 0.5 mark from Assgn 17 if they have not submitted the Intermediate file (unless excused by us)</u>

Assignment 17 (Total Marks = 10):

- Output: 2 marks (1 marks each binary)
    - Test Case 1: Create a 3 x 4 matrix with all positive integers, each row sorted, and 2nd row with all integers less than any in 1st row.
    - Test case 2: Same, but with all negative integers
    - TAs should create the inputs and outputs. Same input for all.
- Code: 8 marks
    - SortMatrix Function: 5 marks
        - Copying to and from 1-d array correctly: 2 marks
        - Sorting: 3 marks
            - Deduct 2 if they do not use merge in some form
        - Note that they have been allowed to use any number of additional arrays (not in assignment slides, changed in class)
        - You may get variations where they have tried to sort directly inside the 2-d array etc., use your judgement then
    - Main function: 3 marks
        - Reading from file, including fopen error check: 2 marks
        - Writing into file: 1 mark

Assignment 18 (Total Marks = 10):

- Output (binary marking: either 1 or 0 for each test case): 2 marks
    - Please create a file with 6 student records in it, with at least 2 sets of 2 records each with same first name (and no record with same first and last name)
    - Test case 1: Give a first name and last name that exists. The input should be from one pair with same first name. Should print the correct record.
    - Test case 2: Give a first name that does not exist but last name that exists. Should print a message saying not found.
    - TAs should create the input and output. Same for all.
- Code: 8 marks
    - Main function: 2 marks
        - Counting no. of structures in file: 1 mark
        - Reading into array of structures malloc'ed: 1 mark
        - Give 0 in this part if they have just assumed affixed constant no. of structures
    - SortStudents function: 3 marks
        - Sorting by first name: 2 marks
            - They can use any sorting method

    - Handling tie-break with last name during sorting: 1 marks
- SearchStudents function: 3 marks
    - Searching by first name: 2 marks
    - Once first name is found, handling correct last name: 1 mark
        - To find the proper last name, they should do linear search on BOTH sides of the index found until that first name does not match
    - Deduct 2 if they have not used binary search for sorting by first name. So they get 1 in this part if they have just done a plain linear search matching first name and last name