COP 3502

Spring 2025

Programming Assignment 4

Student Grade Analyzer Using Linked List

Now that we've explored different sorting techniques, it's time to apply what you've learned. In this assignment, you will write a C program that:

- Reads student names and exam scores from a file
- Stores them in a **singly linked list**
- Sorts the students by score in **descending order**
- Computes and displays the following statistics:

    1. The list of students sorted by score (high to low)
    2. The **highest** and **lowest** grade
    3. The **median** score
    4. The **class average**
    5. The **top 5 students**

Implementation Restrictions

- Your program must use structs and a linked list to store student data.
- You are free to design the program structure, but you **must not place all your code inside `main()`**. Use proper modular design with separate functions for each task.
- You must use a sorting algorithm that runs in **O(nlogn)** time
- You must verify that none of your functions exceed O(nlogn) runtime. (They may run faster, but not slower than O(nlogn)
- You must read input from a text file, and output the results to the terminal **(stdout)**.

Required Struct

```
typedef struct Student {
    char first[30], last[30];
    int score;
    struct Student* next;
} Student;
```

Input Format

- The program should prompt the user to enter the **input file name**, such as:

```
Enter input file name:
```

- Upon entering a valid file name (`input.txt`), the program should read student data and process it.

- Each line of the input file will follow the format:

```
<LastName> <FirstName> <Score>
```

You may assume:

- No duplicate names or scores
- All names and scores are valid

Sample Input File (`input.txt`)

```
Gupta Anil 96
Ball Adam 90
Fox Brian 87
Davis Ronald 62
Cook Bret 55
Arnold Joshua 67
Singh Jaadu 91
Scot Michael 80
Hudson Jeff 45
Evans Rudy 75
```

Expected Output:

```
Enter input file name: input.txt

Sorted List by Score:
Gupta Anil - 96
Singh Jaadu - 91
Ball Adam - 90
Fox Brian - 87
Scot Michael - 80
Evans Rudy - 75
Arnold Joshua - 67
Davis Ronald - 62
Cook Bret - 55
Hudson Jeff - 45

Highest Score: 96
Lowest Score: 45
Median Score: 77.50
Class Average: 74.80

Top 5 Students:
Gupta Anil - 96
Singh Jaadu - 91
Ball Adam - 90
Fox Brian – 87
Scot Michael - 80
```

What to Submit

You must submit the following via **WebCourses** by the due date:

1. A single C source file named: `grade_analyzer.c`
2. Your code must compile and run on Eustis
3. Your output must match the format shown above
4. Follow the style guide provided in previous assignments to earn full credit

Additional Notes

- You are encouraged to create and test your own input files.
- Your submission will be evaluated using five different test cases.
- You are responsible for ensuring your code handles valid inputs correctly.