Ashley Moreno
Lab 1 Part a Write-Up

Team Members
None

Initial Decisions
For my implementation, I decided to use python as my programming language. I am comfortable with the language and its built-in libraries are ideal for tasks like file handling and data manipulation. The program was developed in a Unix-based environment. I used the text editor Vim and all debugging was done directly in the terminal by running the Python script and checking outputs or errors.

Internal Architecture

- The main structure used was a list of lists (students). Each student entry in the students list represents a line from the students.txt file, where each list contains the student's information (last name, first name, grade, classroom, bus route, GPA, teacher's last name, and teacher's first name). This allowed for easy access to each attribute.
- A dictionary was used in the student_info() function to track the number of students in each grade (key: grade number, value: count of students in that grade). This allowed for easy counting and retrieval of the number of students in each grade.
- Lastly, Python's string handling and formatting methods (such as .format(), concatenation, and str()) were used to construct output messages.

Task Log

All tasks completed by Ashley Moreno

Task Name	**Time to Complete**
Project Design & General Set up	**~ 40 min**
Parsing & Loading Data	**~ 1 hr 15 min**
Implementing Search Commands	**~ 3 hrs**
GPA Calculations	**~ 50 min**
Student Info & Quit	**~ 40 min**

Test Suite & Output	**~ 25 min**
Lab Write-Up	**~ 30 min**
README & Submission	**~ 15 min**

Testing

When: Testing was performed throughout the development process, especially after completing each major functionality (e.g., student search, GPA calculations, etc.).

Who: Ashley Moreno

Duration: Approximately 2.5 hours were likely spent testing and debugging the entire program.

Bugs Found:

- Issues with concatenating non-string values (NoneType, integers, etc.) in functions like find_highest_gpa_in_grade() and find_lowest_gpa_in_grade(). This was resolved by ensuring that all values were explicitly converted to strings using str().
- Input validation problems. The program required that input commands be enclosed in quotations (e.g., 'G 3 H'), otherwise it would raise an EOF or Syntax Error.
- Minor issues with input parsing, such as case-sensitivity and optional commands.

Time to Fix Bugs: 1-2 hours

Final Notes

I was unable to fix the input validation bug. While the program works as expected, having to put quotes around each command is kind of weird. Other than that minor quirk, I believe all else went well.