Writeup: Lab 1b

- Team Name:
  - Team Coby
- List of team members:
  - Coby Hong (no teammates).
- Initial Decisions:
  - Decided to work in Python as most familiar language for me currently.
  - Working on windows PC.
  - Prefer parsing method to allow multiple filters to be used.
- Notes on chosen internal architecture:
  - Developed two dictionaries containing keys that led to specific functions.
    - One dictionary for search functions.
    - One dictionary for print functions.
  - File parsing functions that essentially combine the two provided text files into a single list like lab1a file structure.
  - Main returned item would be a 2-item array consisting of a list of students based on filter applied by user and a print code.
  - General Layout:
    - In the back, the two files are read in. Parse the two files to combine into a single list based on the association / primary key they share (classroom number).
    - Prompt and input option given to user.
    - User would type in their input.
    - This input would be parsed to grab keys to search functions and the data tied to those keys.
    - Based on key, program jumps to search function associated to that key and does appropriate filtering of student list. Returns that list with print formatting code.
    - Returned list is then printed based off the print formatting code.
      - This styling allows for multiple filters to be used in a single search and print on the last key search given by the user.
        - EX: >> G: 4 H:
          - Input is parsed.
          - First key is “G:” with data ‘4’.
          - Student list filtered by grade and returns code 4.
          - Second key is “H:”.
          - Student list currently filtered by grade filtered again for highest GPA and returns code 5.
          - Parsed commands complete and does printing based off code to returned list.
- Task Log:
  - Coby Hong – Coding, Testing, Write-Up. All of it.
  - About 7 hours.
- Notes on testing:
  - BUG #3 found was when user provided no input to a language that required one (EX: “A:”), the output would return “DNE: 0”. This is due to not considering length of parsed input and internal structure of handling single piece commands.
  - BUG #2 found was that filters that are not listed to work together do work in my program due to the general design. Since mine parses, the input and does the search filters from left-to-right, inputs such as “T: KERBS B: 52” would work.
- Final notes:
  - Still unsure if my NR5 implementations are good since as described, that requirement was very vague, and I am honestly bad at interpreting instructions. The change in text file input wasn’t too hard but I could see it being incredibly frustrating if taking other methods, especially when needing to mutate a preexisting list to apply multiple filters.