* Ask user what they want to do with the list of students
* Keep looping until user decides to quit
  + Add
    - Create a new student
    - Ask user for last name and first name of the student
    - Combine both names together to form the name of the student
    - Ask for student ID
    - Ask for amount of tests student took
    - Make a dynamic array of test scores using the amount of tests
    - ask user for score of each test
    - Open student.dat to append data
    - Make a new line and write student’s information in the same format
    - delete memory for test scores
    - close the file
  + Remove
    - Ask user for ID to remove
    - get the number of students from getNumber()
    - make a dynamic array of students using number of students
    - set a boolean flag for match and set to false
    - Open student.dat to read data
    - in a loop, go through each line and store students in the dynamic array using comma delimiter
      * while reading the file, if there is a student id that matches the ID given to remove, set match to true
    - close the file
    - if there was a match
      * open student.dat for overwriting
      * write contents of dynamic array to file only if student ID does not match the removedID
      * Make a new line numStudents -2 times (- removed student and - 1)
    - if not
      * inform user the student ID does not exist
    - delete dynamic arrays of testscores for all students
    - delete dynamic array of students
  + Display
    - get the number of students from getNumber()
    - make a dynamic array of students using number of students
    - Open student.dat to read data
    - in a loop, go through each line and store students in the dynamic array using comma delimiter
    - in another loop, display the dynamic array with formatting for each student
      * make another loop to display testscores for each students
    - delete dynamic arrays of testscores for all students
    - delete dynamic array of students
    - close the file
  + Search
    - Ask user for ID to search
    - create a boolean flag for match and set to false
    - open student.dat for reading
    - create a student pointer to store each student being read
    - in a loop, go through each line and store a student inside student pointer using comma delimiter
      * while reading, if studentID matches the ID being searched for, set match to true and display data of that student with formatting, break out of the loop to stop reading file
    - delete dynamic arrays of testscores for all students
    - delete dynamic array of students
    - close the file
    - if a match did not exist
      * inform user the student ID does not exist
  + Results
    - create a file averages.dat for writing
    - open student.dat for reading
    - make a dynamic array of students using number of students
    - Open student.dat to read data
    - in a loop, go through each line and store students in the dynamic array using comma delimiter
    - in another loop. calculate averages for each student
      * if student has taken all 5 tests, subtract minimum score from sum and decrement number of tests
      * calculate average and write average to averages.dat with 1 decimal place
    - close both files
    - delete dynamic arrays of test scores for all students
    - delete dynamic array of students
  + Quit
    - Exit the program
  + Default
    - if user entered an incorrect choice, inform them and keep looping