BY SUBMITTING THIS FILE TO CARMEN, I CERTIFY THAT I HAVE PERFORMED ALL OF THE WORK TO CREATE THIS FILE AND/OR DETERMINE THE ANSWERS FOUND WITHIN THIS FILE MYSELF WITH NO ASSISTANCE FROM ANY PERSON (OTHER THAN THE INSTRUCTOR OR GRADERS OF THIS COURSE) AND I HAVE STRICTLY ADHERED TO THE TENURES OF THE OHIO STATE UNIVERSITY'S ACADEMIC INTEGRITY POLICY.

THIS IS THE README FILE FOR LAB 4

Your Name: Yi Chen

Total amount of time it took for me to complete this lab: 42 hours

There were many concerns of the lab. When I try to insert nodes into the linked list. The nodes were added in decreasing ID number instead of increasing ID number. When reading in the student names, some garbage values were printed. There were issues with cases where names contain more than one word. When calculating averages, I wasn't sure whether to do by cumulative score or by individual scores for each category. So, I had to put the files into the Excel spreadsheet to test it. I didn't know whether I should add the n/a or not. There were minor differences in the sample output value and my output value. There were some mistakes of not resetting the temporary variables to 0 when calculating each student which caused very big numbers printed out. When printing the header, I realized that the length of the characters was 15 instead of 20 when examining the printHeader file. The #includes were very confusing causing my code not to compile. I didn't understand that I should do a #include for the .c files for each function or put function headers in the .h file and #include "lab4.h". When I set option to 10, the function for option 10 doesn't execute which caused the output file on disk not to be update So, I moved it from the loop to outside the loop through process options. Initially, I processed the options in the main method. Until one class, I noticed that they should go into a separate .c file. I wasn't sure whether I should dynamically allocate an array of characters in the student name in option 2. I got segmentation faults if I used a statically declared array. I was confused with which command to compile the program. I initially thought it was gcc -ansi -pedantic -g -o lab4 lab4main.c, but I found out that it should be gcc -ansi -pedantic -g -o lab4 *.c *.h. There was a time when I mistyped the command as gcc -ansi -pedantic -g -o *.c *.h and got "undefined reference to allocate_node" and the file got deleted. When I looked at the sample output, I realized that the program should search for the whole student name instead of the last name. There was a time when I run, I got an invalid pointer to free. This was happened when I called free(argv[1]) when no allocated storage. When printing the student data, I was confused on how to properly format it and get the data to fit into one column. I wasn't sure whether I should dynamically allocate the array of pointers to character pointers for the categories. In option 2, I tried to match by the last name of each student, but I wasn't able to. I couldn't find which string library function in C could read from the beginning of a string to the first occurrence of the substring. Also, when I directly modify the structure members, it didn't work well, so I had to store the value into a temporary variable and then transfer it into the member.

I used gdb to test the command line arguments. When I break it into main, argc was 1. However, when I ran the program, argc was 3. I used gdb when calculating the average values for the categories and compared it with the Excel output. When reading in student data, I used gdb for the values of the student data. I used static declared array and dynamically allocated array for the categories. However, when I ran gdb, it is giving me a segmentation fault in both ways. When I ran the program, there were no segmentation faults in my program.