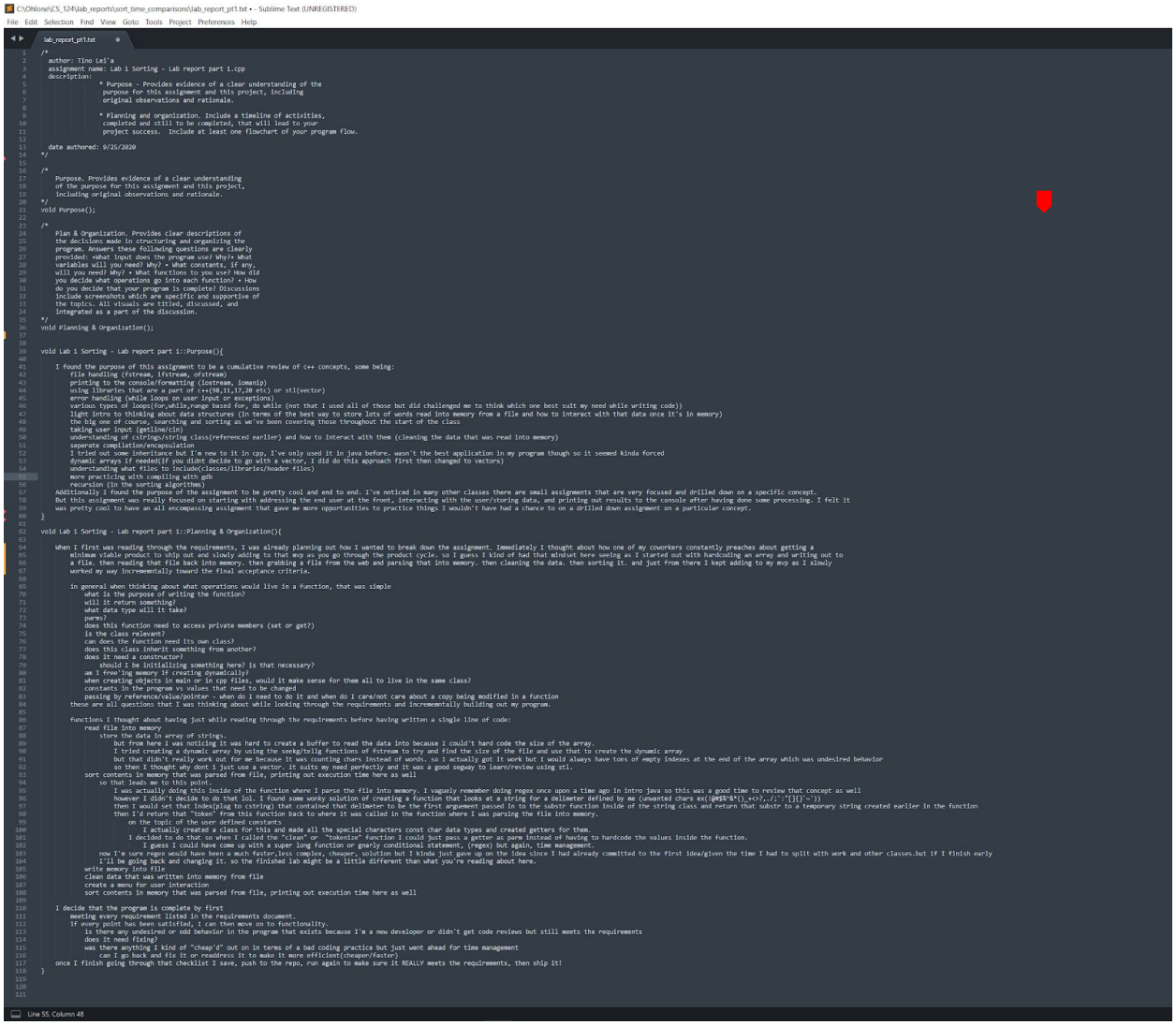

// pdf || google doc were the only accepted file types so I just copy pasted what I wrote in my
// .txt file here
// ps. I hope you get a kick out of this format of the lab write up. purely did it to keep it humorous and fun. if it's a pain for you please let me know and I can resubmit

/*
  author: Tino Lei'a
  assignment name: Lab 1 Sorting - Lab report part 1.cpp
  description:

* Purpose - Provides evidence of a clear understanding of the purpose for this assignment and this project, including original observations and rationale.

* Planning and organization. Include a timeline of activities, completed and still to be completed, that will lead to your project success. Include at least one flowchart of your program flow.

```
  date authored: 9/25/2020
*/

/*
        Purpose. Provides evidence of a clear understanding
        of the purpose for this assignment and this project,
        including original observations and rationale.
*/
void Purpose();

/*
        Plan & Organization. Provides clear descriptions of
        the decisions made in structuring and organizing the
        program. Answers these following questions are clearly
        provided: •What input does the program use? Why?• What
        variables will you need? Why? • What constants, if any,
        will you need? Why? • What functions to you use? How did
        you decide what operations go into each function? • How
        do you decide that your program is complete? Discussions
        include screenshots which are specific and supportive of
        the topics. All visuals are titled, discussed, and
        integrated as a part of the discussion.
*/
void Planning & Organization();


void Lab 1 Sorting - Lab report part 1::Purpose(){

        I found the purpose of this assignment to be a cumulative review of c++ concepts, some being:
                file handling (fstream, ifstream, ofstream)
                printing to the console/formatting (iostream, iomanip)
                using libraries that are a part of c++(98,11,17,20 etc) or stl(vector)
                error handling (while loops on user input or exceptions)
                various types of loops(for,while,range based for, do while (not that I used all of those but did challenged me to think which one best suit my need while writing code))
                light intro to thinking about data structures (in terms of the best way to store lots of words read into memory from a file and how to interact with that data once it's in memory)
                the big one of course, searching and sorting as we've been covering those throughout the start of the class
                taking user input (getline/cin)
                understanding of cstrings/string class(referenced earlier) and how to interact with them (cleaning the data that was read into memory)
```

seperate compilation/encapsulation

I tried out some inheritance but I'm new to it in cpp, I've only used it in java before. wasn't the best application in my program though so it seemed kinda forced

dynamic arrays if needed(if you didnt decide to go with a vector, I did do this approach first then changed to vectors)

understanding what files to include(classes/libraries/header files)

more practicing with compiling with gdb

recursion (in the sorting algorithms)

Additionally I found the purpose of the assignment to be pretty cool and end to end. I've noticed in many other classes there are small assignments that are very focused and drilled down on a specific concept.

But this assignment was really focused on starting with addressing the end user at the front, interacting with the user/storing data, and printing out results to the console after having done some processing. I felt it

was pretty cool to have an all encompassing assignment that gave me more opportunities to practice things I wouldn't have had a chance to on a drilled down assignment on a particular concept.

}

void Lab 1 Sorting - Lab report part 1::Planning & Organization(){

When I first was reading through the requirements, I was already planning out how I wanted to break down the assignment. Immediately I thought about how one of my coworkers constantly preaches about getting a minimum viable product to ship out and slowly adding to that mvp as you go through the product cycle. so I guess I kind of had that mindset here seeing as I started out with hardcoding an array and writing out to a file. then reading that file back into memory. then grabbing a file from the web and parsing that into memory. then cleaning the data. then sorting it. and just from there I kept adding to my mvp as I slowly worked my way incrememntally toward the final acceptance criteria.

in general when thinking about what operations would live in a function, that was simple

what is the purpose of writing the function?
will it return something?
what data type will it take?
parms?
does this function need to access private members (set or get?)
is the class relevant?
can does the function need its own class?
does this class inherit something from another?
does it need a constructor?
should I be initializing something here? is that necessary?

am I free'ing memory if creating dynamically?

when creating objects in main or in cpp files, would it make sense for them all to live in the same class?

constants in the program vs values that need to be changed

passing by reference/value/pointer - when do I need to do it and when do I care/not care about a copy being modified in a function

these are all questions that I was thinking about while looking through the requirements and incrememntally building out my program.

functions I thought about having just while reading through the requirements before having written a single line of code:

read file into memory

store the data in array of strings.

but from here I was noticing it was hard to create a buffer to read the data into because I could't hard code the size of the array.

I tried creating a dynamic array by using the seekg/tellg functions of fstream to try and find the size of the file and use that to create the dynamic array

but that didn't really work out for me because it was counting chars instead of words. so I actually got it work but I would always have tons of empty indexes at the end of the array which was undesired behavior

so then I thought why dont i just use a vector. it suits my need perfectly and it was a good segway to learn/review using stl.

sort contents in memory that was parsed from file, printing out execution time here as well

so that leads me to this point.

I was actually doing this inside of the function where I parse the file into memory. I vaguely remember doing regex once upon a time ago in intro java so this was a good time to review that concept as well

however I didn't decide to do that lol. I found some wonky solution of creating a function that looks at a string for a delimeter defined by me (unwanted chars ex(!@#$%^&*()_+<>?,./;':"[]{}`~'))

then I would set that index(plug to cstring) that contained that delimeter to be the first arguement passed in to the substr function inside of the string class and return that substr to a temporary string created earlier in the function

then I'd return that "token" from this function back to where it was called in the function where I was parsing the file into memory.

on the topic of the user defined constants

I actually created a class for this and made all the special characters const char data types and created getters for them.

I decided to do that so when I called the "clean" or "tokenize" function I could just pass a getter as parm instead of having to hardcode the values inside the function.

I guess I could have come up with a super long function or gnarly conditional statement, (regex) but again, time management.

now I'm sure regex would have been a much faster,less complex, cheaper, solution but I kinda just gave up on the idea since I had already committed to the first idea/given the time I had to split with work and other classes.but if I finish early

I'll be going back and changing it. so the finished lab might be a little different than what you're reading about here.

write memory into file

clean data that was written into memory from file

create a menu for user interaction

sort contents in memory that was parsed from file, printing out execution time here as well

I decide that the program is complete by first

meeting every requirement listed in the requirements document.

if every point has been satisfied, I can then move on to functionality.

is there any undesired or odd behavior in the program that exists because I'm a new developer or didn't get code reviews but still meets the requirements

does it need fixing?

was there anything I kind of "cheap'd" out on in terms of a bad coding practice but just went ahead for time management

can I go back and fix it or readdress it to make it more efficient(cheaper/faster)

once I finish going through that checklist I save, push to the repo, run again to make sure it REALLY meets the requirements, then ship it!

}