Purpose

The purpose of this program is to explore dynamic memory and practice pointers through building a custom string class. The program must adhere to provided specifications and pass the provided driver file.

Design

I followed the specifications. I implemented the private methods to modify the buffer. Then, I implemented the constructors and used initializer lists to set the pointers to null and m\_size to 0. I then used the buffer methods to import any data passed to the constructors. Implementing the << operator was easy as the storage for the string was specified to be in a cstring format. The size and length functions are essentially the same, as one includes the NUL terminator while the other does not. The destructor simply deallocated the buffer using the private deallocation function. I used m\_size for both and decremented by one in order to avoid running strlen on each function call, a relatively expensive operation. The equality operator simply runs strcmp on the buffers of both strings, as both buffers are valid cstrings. The assignment operator essentially emulates the copy constructor as it copies most of its functionality. The addition operator was fairly complex. It allocates a new buffer with room for both, copies the first string, then concatenates the second. The indexing operators simply index into the buffer.

Problems/Changes

The String performs a deep copy on each copy construction, so it is extremely expensive. There is no way to fix this while adhering to the provided specifications. I also would like to have made buffer reallocation copy the old data, but this is also not possible while following the provided specifications. I would also have liked to make the string to be backed by a vector-like storage for easier interactions, but this is not possible with the given specifications.