I combined part 1 and part 2 into a single program (file submitted alongside this), The file actions work a lot better than on the last assignment, I think it’s because I opened and closed the file for every read/write operation instead of just opening it and leaving it open until the program was finished.

The stack function works exactly as it should, printing an exact reverse copy of the original file.

The qeue function works a little differently than the explicit instruction (to prioritize all vowels to the head of the qeue). What I tried to do was build a priority qeue, which operated on the same fifo principal (rather than just creating a stack of all the vowels prepended to the qeue), that lead into a qeue of consonants.  
--- there’s only one issue that isn’t debugged; If the first letter in the original file is a consonant, it will remain the first letter when printed- followed by all the vowels, then the rest of the consonants. (I tried addressing it, in lines 181-189, but it didn’t solve the problem. Maybe there was a line in the consonant qeue block that was interfering.) Otherwise it worked as I expected

Note: the DSU compiler continued to start the program in the middle of the main function—after running through the first subroutine/loop it would run the program as expected.

**PART 3: Examples of real world stacks/qeues**

**Stack:** stack of plates, every cleaned plate gets piled on top of the others, and when you need one you take off the top of the stack- not the bottom- so the first one on the stack is the last one to leave it (hence filo acronym).

**Qeue:** the line at any pay business’ register (movie theatre, McD’s, best buy, Qdoba, etc. anywhere with a line) is a qeue, each person/group starts at the back of the line, and leave after being helped (at the front). As such the fist person in the qeue is the first one out (hence the fifo acronym).