This project went by a little faster than the last few mostly because implementing templates didn't really require any major changes to the code structure itself. I mainly just needed to add the template keyword to everything and change the data type to the template parameters but that was pretty much it.

The stack behavior wasn't all that different from the queue, and thus a lot of my functions performed basically the same as that of the queue. The main difference between the two was probably between the copy constructor and the = operator. In this project I made use of all the other functions in the class, like empty, pop, and push, as opposed to the other ones where each function was more or less self contained and didn't rely on others to do its job. This code reuse made the overall project a lot shorter, and I really should have thought of this before. Besides that, I made a special copy function that both the copy constructor and the = operator made use of. It works much like the destructors from previous projects, making use of recursion to find the back of the queue and push items from back to front. There were other ways of doing the copy operation, but I felt like this was the most efficient, both in terms of code length and performance.

The driver code was pretty much just copy-pasted from my last project, modified to work with the new templated class. In my driver code, I tested both classes with int and char types, although I'm pretty sure using the DataType class would also work as well.