

CS162 – Module 2 Testing

Daniel Beyer

In order to test the structures in my Stack and Queue classes, I created a `display()` function that would output the node contents for testing purposes after all values had been inputted by the user but before the `remove()` function had been called. This would allow me to make sure the nodes were being created and ordered correctly. I could then call the `remove()` function and determine if the nodes were being “popped” off correctly. The results are in the table below:

Test Case	Input Values	Driver Functions	Expected Outcome	Observed Outcome
User enters sequence of characters into Stack	“a”, “b”, “c”, “d”	s.display()	“d” “c” “b” “a”	“d” “c” “b” “a”
User enters sequence of characters into Stack	“a”, “b”, “c”, “d”	s.remove()	“d” “c” “b” “a”	“d” “c” “b” “a”
User enters sequence of characters into Queue	“a”, “b”, “c”, “d”	q.display()	“a” “b” “c” “d”	“a” “b” “c” “d”
User enters sequence of characters into Queue	“a”, “b”, “c”, “d”	q.remove()	“a” “b” “c” “d”	“a” “b” “c” “d”

Based on the results of this testing, I determined the nodes were being added to the Stack and Queue objects correctly. Since my `remove()` functions displayed the node values in the specified order, I determined the nodes were being removed in the correct manner as well.

To test for memory leaks, I ran my program using Valgrind on FLIP using the command line “`valgrind –leak-check=yes mod2`”. The results were 0 errors and all heap blocks freed – no leaks possible. This allowed me to determine I was deallocating the memory correctly.