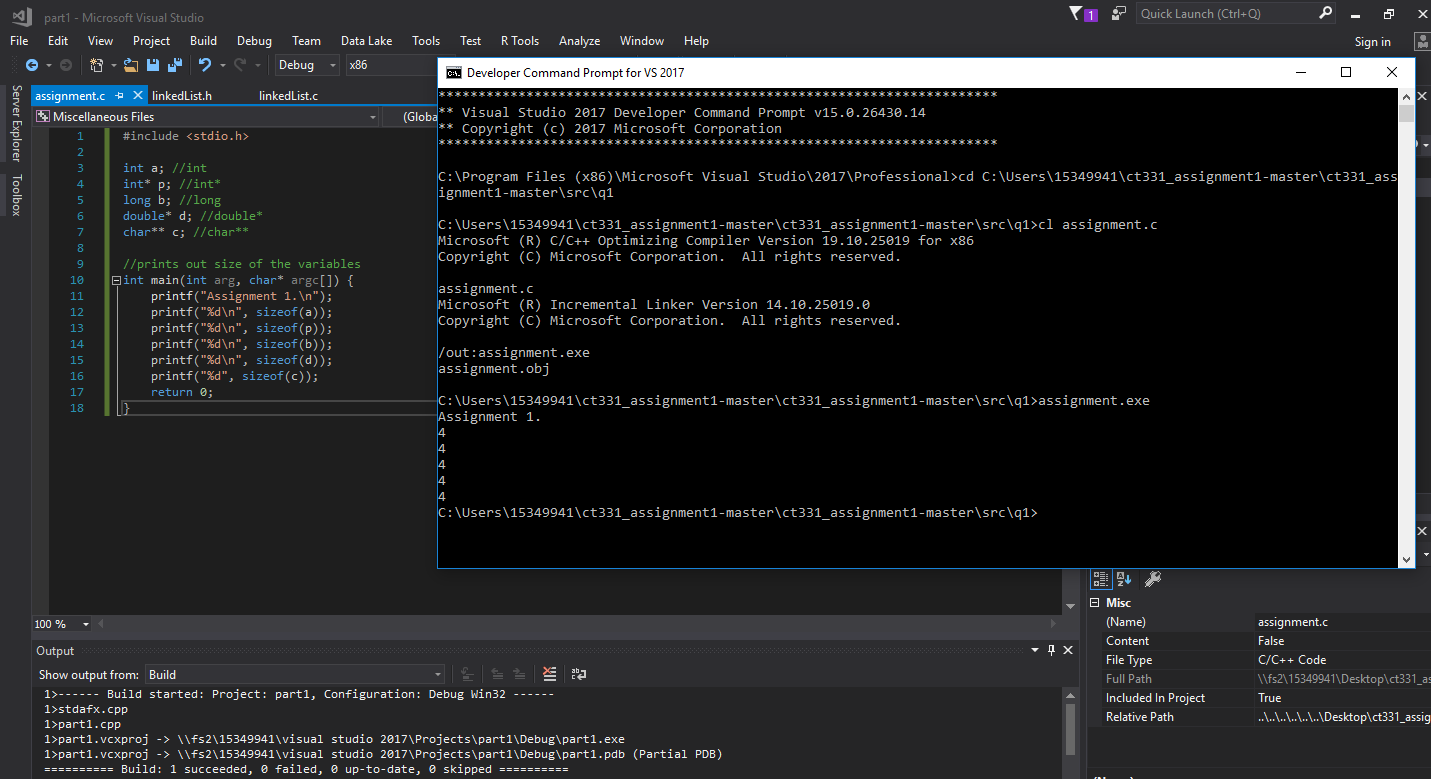
Marc blain 15349941

<https://github.com/MarcBlain/ct331_assignment1>

Hi Aiden we talked earlier in the 9am lecture and you advised me to resubmit the assignment after we had talked, I have changed the GitHub format and now you can access the src better and I have added the word document to the repository. I have not done any tests for the code.

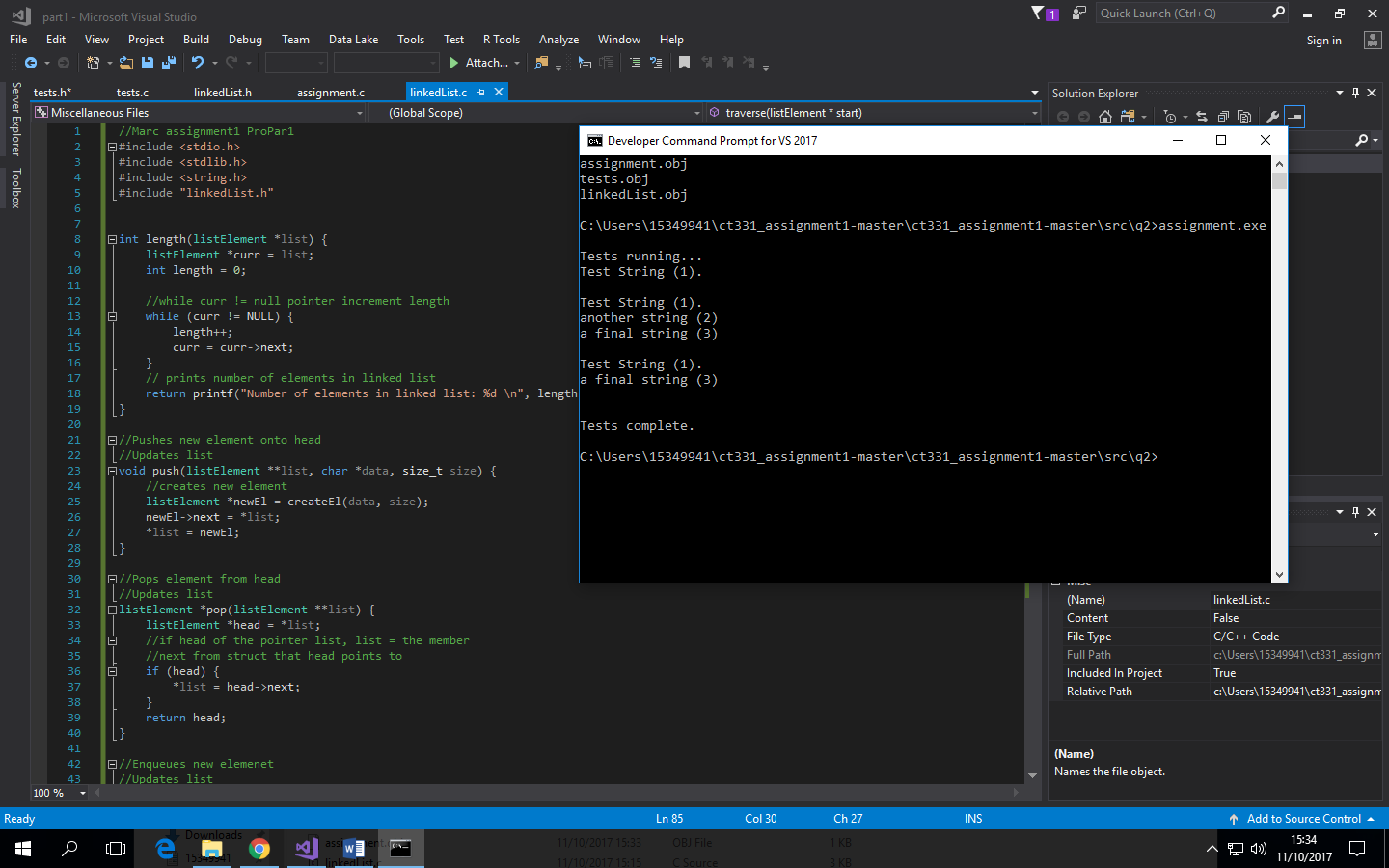
Part 1



As seen in the screenshot the size of each variable is 4. The amount of memory needed to store all of the different pointer types ended up being identical, even though the storage needed for the data they may point to is different.

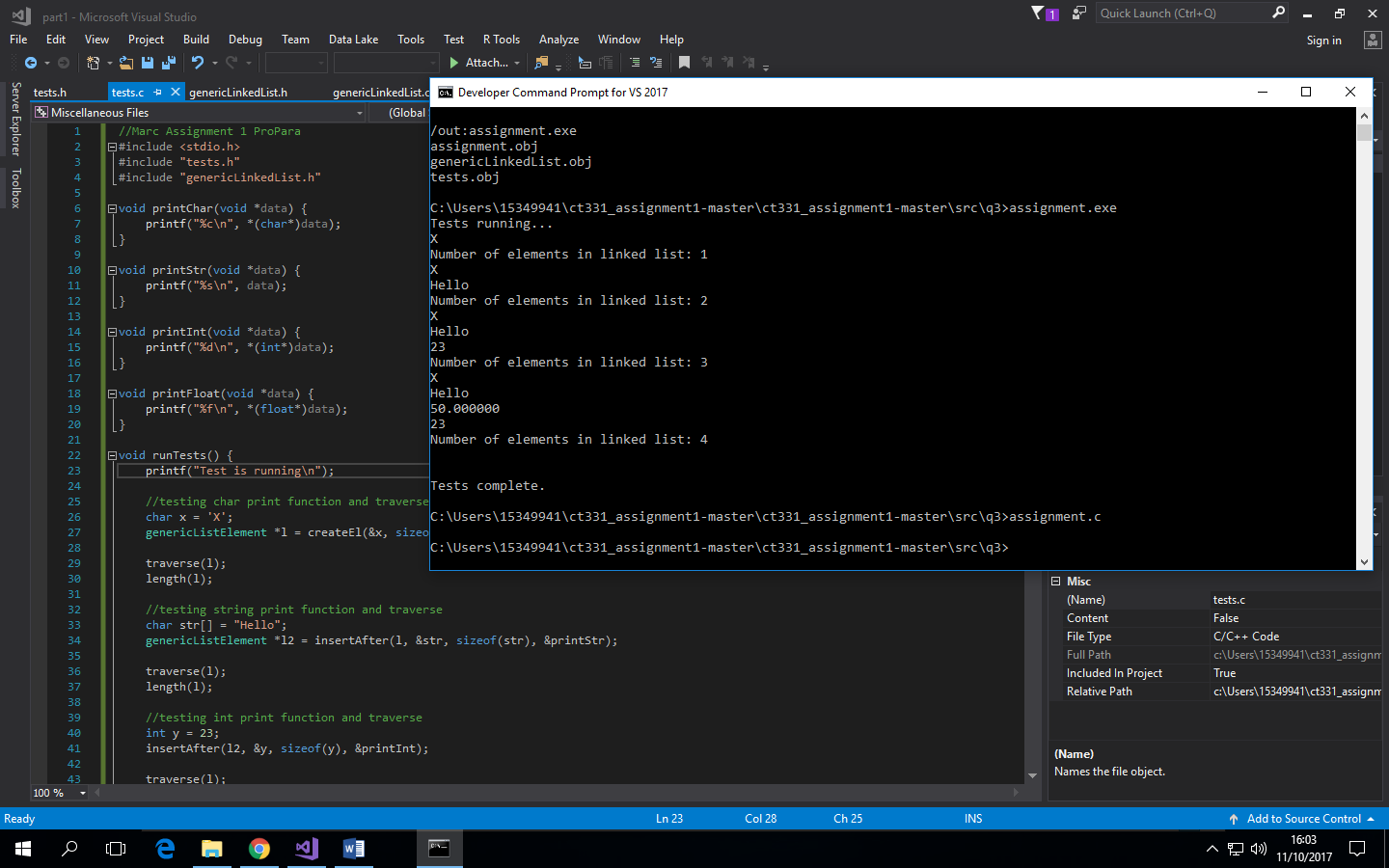
The pointers are pointing to the size of the address and not the actual variables

**Part 2**



Part 2 completed and it tests the add and remove string methods with the push and pull also

**Part 3**



Part 3 completed as it adds elements in each linked list and finished with 4 different variables s

**Part 4**

* When traversing a single linked list from the tail to the head there is no way to increment backwards.
* If the length of the list is known, each piece of data or node could be stored in an array as it is passed, then the array iterated backwards to get a final result.
* To improve the memory usage of a reversed traversal, the program would have to store a “previous” pointer within each node (doubly-linked list) as well as maintaining a pointer to the tail node of the list.
* Using a double linked list lets us go forwards and backwards in the list as to just one direction.