In this assignment, I implemented a stack-based shopping list using a linked list. Each shopping item was stored in a Node object containing the item data and a pointer to the next node. I implemented the standard stack operations such as push, pop, and peek. These operations correspond to inserting an item at the head, removing the top item, and retrieving the top item without removal, respectively. Additionally, I implemented recursive methods to print the stack from top to bottom and from bottom to top. This demonstrated how recursion can be used to traverse a data structure in multiple ways. The program output shows that the stack maintains the correct Last-In-First-Out (LIFO) behavior. The most recently inserted item appears at the top, is returned by the peek operation, and is removed first when performing a pop operation. The recursive printing methods display the stack in both directions, confirming that the recursion logic is implemented correctly.

Through this exercise, I deepened my understanding of how linked lists work as a dynamic data structure. Such as managing head pointers during insertion and removal. I also reinforced the practical use of recursion for traversing non-linear structures and observed firsthand how the call stack operates when printing items from bottom to top. While I did not encounter significant difficulties in completing the assignment, the exercise highlighted the importance of careful pointer management and methodical implementation to ensure that stack operations behave as expected and that the printed output matches the required format. From this assignment, I was able to strengthen my knowledge of linked list-based stacks, recursion, and object-oriented programming in Python.