Charlie Say

CS 260

Assignment 2 (Design)

Based on what we know about linked lists, stacks, and queues, design a queue data structure:

What functions are we likely to need for a queue to function like the one discussed in class?

- Add Data (Inserting data into a stack - Push)

- Remove Data (Deleting item from the stack - Pop)

- View Data in the stack

- Top() - returns last item pushed to the stack

- is-empty() - True if no more items can be popped and there is no top item

- is-full() - True if no more items can be pushed

- get-size() - get length of the stack

What variables will we need at the structure level to track for our queue to function properly?

- class Stack {}

- data type pop() (remove from the top of the stack and return the value)

- length = 0;

- char\_node \*new\_node = new char\_node()

- top = new\_node

- new\_node.value - new\_letter

- new\_node.next = NULL

Based on what we know about linked lists design a list data structure that allows us to add or remove values at a given location in the list (instead of the top of a stack or the front or back of a queue):

What functions are we likely to need for a list to function like this?

- Push

- Pop

- Peek

What variables will we need at the structure level to track for this list to function properly?

- head reference

- if head needs to be removed

- find previous node of the node to be deleted or added

- free memory/node from linked list