a. A description of the objectives/concepts explored in this assignment including why you think they are important to this course and a career in CS and/or Engineering.

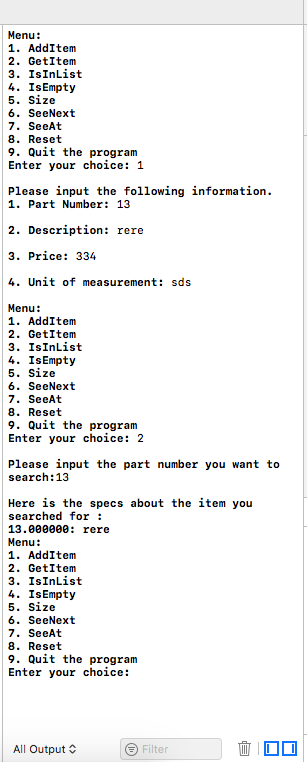
Linked list is explored in this chapter of lab. This is a very important concept in C++ and data structure as a whole. Since other data structure types have their own flaws, for example, array being unable to change its size, and vector being really inefficient in inserting an element in the middle of it, Linked List is the best of both worlds. The programmer can dynamically insert an element in the middle of the list without much trouble using pointers, and its size can go on forever, as it’s only a chain of elements with a pointer pointing to null at the end. Linked List and its pointers can be used as a dynamic implementation of very useful data structures, such as Stacks, Queues, and Binary Tree,...

b. The sections from each task indicated to be included in the lab report.

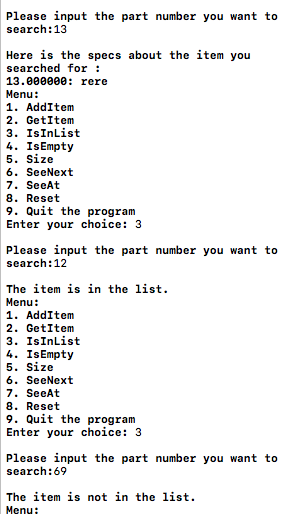
Task 1:

Task 2 & 3:

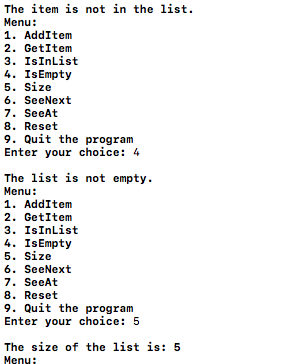
1. AddItems(): Prompt people to input partnumbers, description, price and unit of measure and pass it into a constructor of Item, append it to the list.
2. GetItems(): Compare the partnumber being pass into the function, and remove that item from the list, return that item with parts information



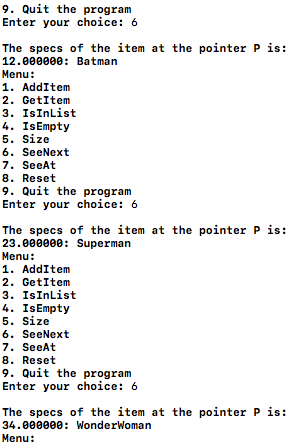
1. IsInList(): Check to see if the item with the partnumber is in the list



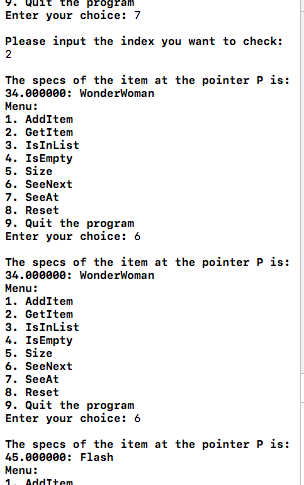
1. IsEmpty() check if the list is empty
2. Size() check the size of the list



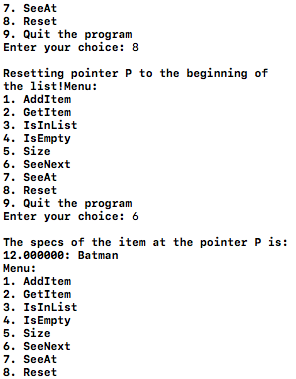
1. SeeNext() Return the item information at the location of pointer P, then move pointer P to the next point on the list.

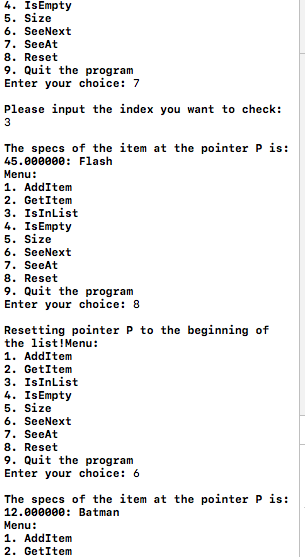


1. seeAt(int) will return the item at the index, and set pointer P to that index



1. reset() set the pointer P at head of the linked list





1. Will quit the loop completely

\*DISCLAIMER:

* I came to Tuesday lab to discuss the wording of the problem, as I do not understand what is asked of me for GetItems(), SeeAt() and SeeNext(). I read that we must return a Node pointer, however, in doing so we wouldn’t be able to call the function from outside of the class, thus we must create other functions to access it. I asked him and he said it’s OK to return the T type object. I would agree that it is more convenient. I have created the node pointer member functions, and comment them out.
* Only a few my exception-throwing works. Since I didn’t return pointers but value, I had to throw errors, and a lot of them for some reason didn’t work, I have not found a way to make them work.
* Ayumi and we have helped each other a lot. I have found her method of creating a pointer P the best solution to the problem, I could not find a better way around it, so I also use such method.
* It wasn’t clear what we’re supposed to be overloading operators on, so I overload operators in all objects: Item class, LinkedList class and Node struct.
* I am aware that some functions of mine don’t work right in Visual Studio. Let me know and I can demonstrate them in Xcode.