1. insertEndMethod - O(n)
   insertIndexMethod – O(n)
   deleteIndexMethod – O(n)
   editIndexMethod – O(n)
   printMethod – O(n)
   searchMethod -O(n)
   main-loop until quit is entered
2. I think the linked list for a line editor is a poor choice when it comes to the methods needed but a good choice when it comes to the storage of the information. A lot of the commands given revolve around a specific index so something like an array or a vector would be better. Instead of O(n) for insert, delete, and edit they would have a complexity of O (1) if a data structure that kept track of indices was used. However, the advantage of using a list is that it doesn't have to have a fixed size, so you can easily add to the end (adding more lines) without having to copy over all the values before that.
3. The first thing I learned is that I needed to dedicate more time spread out over multiple sessions of working on these projects. After looking at a program for so long things stop making sense and your ability to follow what the code is doing starts to fall. It helps to have a break while working on these projects so you get a fresh look at what possible problems could be. As far as this project specifically if I were to do this project again I would spend more time in the logic used in the main method. I would implement a method that broke up the string the user gave first and used that information to decide which command to run instead of doing it differently each time depending on what command the user gave. I would also include more fail safes to prevent the user from crashing the code based on accidental incorrect inputs. Finally I would look into a way of keeping track of where the current end of the list is so to reduce some the complexity in the method of insertEnd(). It would also in general open up more options on how I could better organize my list traversals so I don't have to use multiple temporary variables.