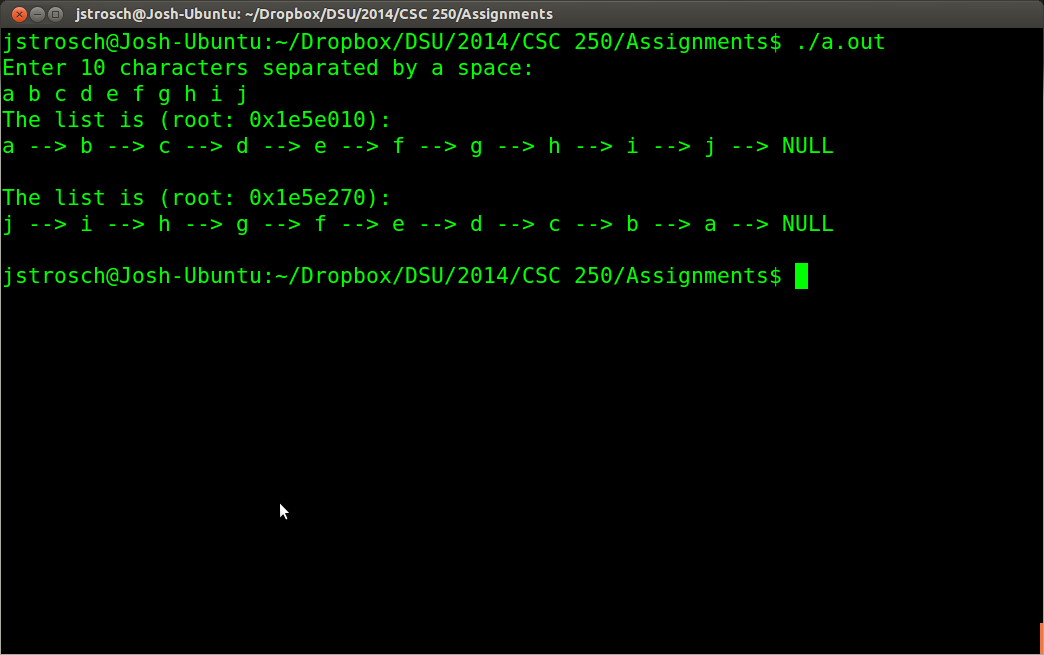
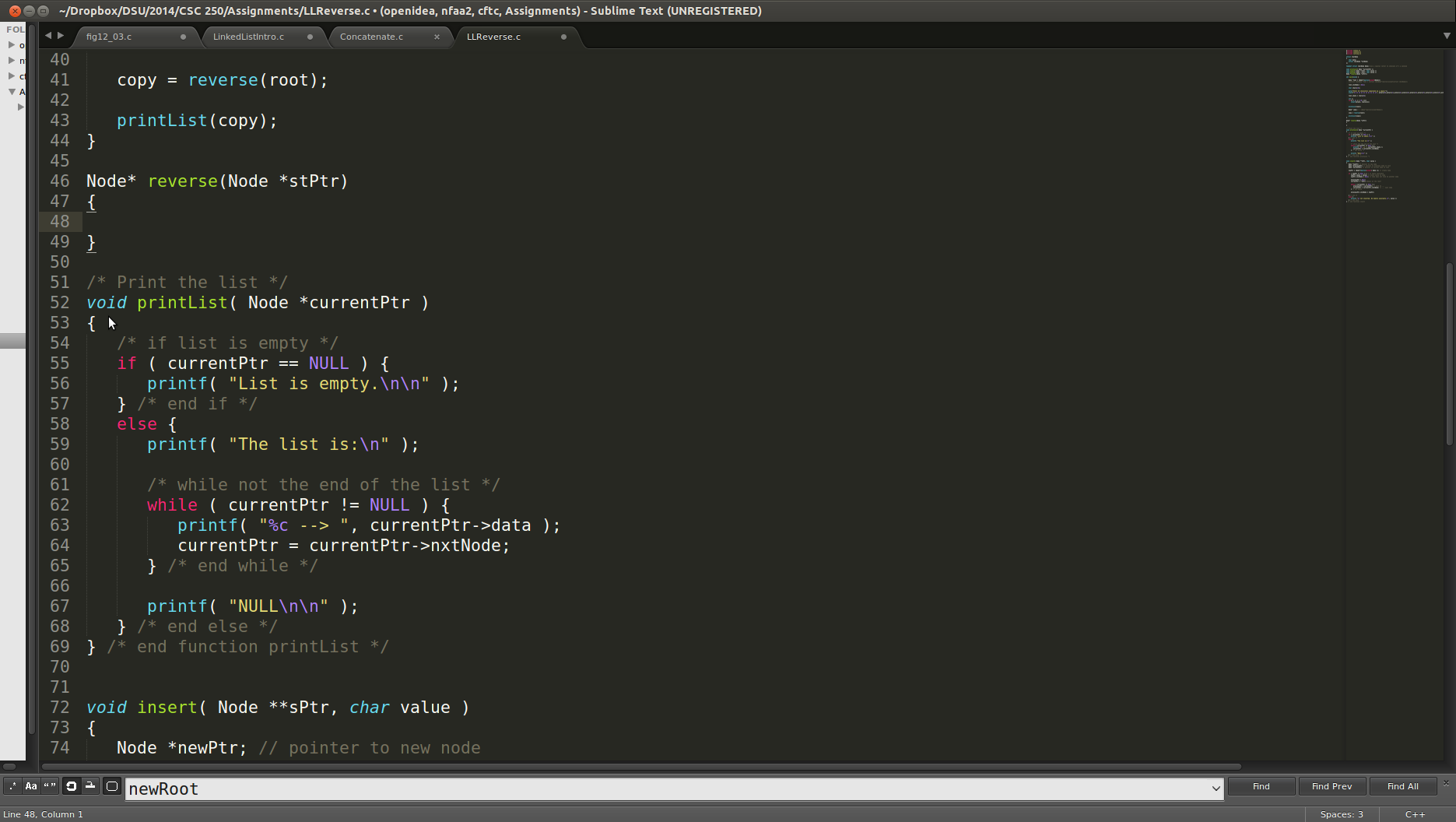
Using the code that we’ve developed in class for linked lists, modify your main function to accept 10 characters from STDIN and create a linked list with those characters (so your nodes will have a char member). Then, add an additional function called **reverse.** The purpose of the function will be to create a copy of the linked list with the nodes reversed. Finally, print off the original linked list as well as the reversed copy. The reverse function will have the following properties:

* The return type is: Node\*
* Has one argument: Node\* stPtr
* Make sure to return the correct memory address to assign to a pointer which is the copy of the list in reverse
* Nothing is printed in this function, its sole purpose is to create a copy of the linked list in reverse order and then return the starting memory address
* This function can be completed with approximately 17 lines of code
* You only need one loop - you’ll need to walk the original linked list to the end (ie nxtNode is NULL)

You’ll use similar logic as we did in the insert function, the only difference is that instead of creating a new node to append (or insert), you’ll create a new node for a new linked list.

Sample Output – Notice each linked list starts with a different memory address:





Submit your program file (linkedLists.c) to the appropriate d2l dropbox.

**REMEMBER: IF YOUR PROGRAM DOES NOT COMPILE WHEN I DOWNLOAD AND COMPILE IT, YOU WILL LOSE ½ OF THE GRADE POSSIBLE.**