# Week 8 Lab: Double Linked List

## This Week

This week, we will be starting the stack backlog. Next week you will demonstrate stories 1-4 in the Stack Backlog: the user interface, simple stack operations.

Implementing stacks are easy after doing the linked lists: you always add and delete from the head of the list (We call it the top of the stack).

Since stacks will be easy for you now that you have done linked lists, this lab will introduce a new user interface: the command line interface that Unix uses. In this interface, instead of starting the program and entering information from a menu to control the program, you call the program with a set of parameters. For example, you can call the `ls` program with the parameters `-l` or `-a` and it will do different things. All of the things you type on a line with a program are available to that program in the parameters to the `main` function. You remember that you can define the main function by `int main(int argc, char *argv[]);`. When you call main this way, `argc` is the number of strings, `char *`, in the array `argv`. You can thing of `argc` as the length parameter to all of the array functions you wrote and `argv` as the array itself.

In the first story, you just print out the array `argv`; in the second, you print out a list of the parameters to the function when you enter `stack -h`. This is the standard way of implementing the inteface. Most of these stories use what you learned in implementing the arrays backlog.

## Demo

You will be demoing the doubly linked list functions from the linked list. This completes the linked list backlog.

## Story 7-12 (15 Marks each; 90 marks total)

Run “make clean,” “make test” and “make.” Execute “./tst” to see if the basic tests pass. Execute “./ll” to make sure that the rest of the program runs.

1. Basics (5 Marks)
    1. Formatting is correct (2 Marks)
    2. Compiles without errors or warnings (3 Marks)
2. Correctness  (10 Marks)
    1. Execute ./tst. The program must use linked lists appropriately to gain the marks.
    2. No other errors (5 Marks)