Homework 8: Serialization

**Assigned**: May 2nd, 2018 **Due**: May 9th, 2018

Make a subdirectory "hw8" of your cs240 folder for this assignment. Copy the starter files from /courses/cs240/s18/kamaral/GROUP/hw8 to your hw8 subdirectory.

**Part 1: Storing a Linked List**

In the file single.c, there are two functions. One of them, storell, is already written for you.

void storell(struct link \*root);

The function takes a root node and stores it to a file single.bin as binary data. It does so following the method we showed in the slides, by storing the list one element at a time in sequential format.

void restorell(struct link \*\*rootptr);

Your job is to write the function restorell which reads the elements from single.bin one at a time and produces links for them. The first link should be assigned to the root variable whose address is given as an argument, and then each other link should be added in sequence.

To reiterate, the argument to restorell is a pointer to a pointer. It is the address of the pointer variable where we should store the address of the first link.

To make things easy on yourself, keep track of the last link in the list using another variable, for example a last pointer. This way, you don't have to keep searching for the last element of the list whenever you want to add the next element.

Keep track of when the fread function is out of elements, as that's when you have read the entire linked list and should terminate the function. Make sure that the last element of the linked list points to NULL and not junk.

**Part 2: Doubly-Linked Lists**

Doubly-Linked Lists are linked lists in which each node in the list also points to its previous element.

In the file double.c, there are two functions. One of them, storell, is already written for you.

void storedll(struct doublelink \*root);

Note that it is nearly identical to the singly-linked case. This is because the data structure still describes a list of data, so we only need to iterate it in one direction: the same direction as the singly-linked list. This function stores the list content in double.bin.

void restoredll(struct doublelink \*\*rootptr);

Your job is to write restoredll which reads the content of double.bin and populates a doubly-linked list in memory. It should be possible to iterate over the list forwards (following the next members) and backwards (following the prev members). Doing so should result in the reversed order of the other.

**Final Remarks**

A makefile has been provided for you which creates a program serial. The serial program creates linked lists of both varieties and tests both of your restoreX functions that you've written. Do no modify any files other than the ones mentioned above and only the modify in those files the functions you're asked to write. You do not need to include any additional global variables to make those functions work. As always, include a typescript showing.

***Personal Exercise***

Once you're done, create a copy of the homework folder in your home directory to avoid messing up your homework submission. Look at what happens when you change the 72nd line of the main.c from:

storedll(droot);

to

storell((stuct link\*) droot);

Does the program still store the doubly-linked list as intended? Why or why not?