CSCD 240
Lab 18

One of the most important data structures that will be used is a Linked List. It is important that we develop a very generic singlularly linked list (no dummy head) that we can use during the course of the quarter.

<u>Node Specifics</u>

- void pointer for data
- next pointer
- Declared in the header file named linkedList.h – provided and can't be changed

<u>LinkedList Specifics</u>

- Node pointer for head (**NO** dummy head node)
- int size
- Declared in the header file named linkedList.h – provided and can't be changed

<u>Specific Functions</u>

- buildList – takes the linked list, the FILE pointer, the total records in the file, and a function pointer to build the appropriate data structure for the type as a parameters. It reads each word from the file and adds the word as part of a word structure to the linked list in the order read in. The memory for each Node will be dynamically allocated and the memory for the data structure will be handled by the build function pointer. This function will run until the end of file is reached.– Located in listUtils.h & listUtils.c – It is not required for every list so this is why it resides in listUtils. – uses addLast

- sort
  - Sort by the char * word similar to a dictionary.
  - Located in listUtils.h & listUtils.c – It is not required for every list so this is why it resides in listUtils.

- printList – if the list is empty prints "Empty List" otherwise prints the list comma separated – in our case it prints the word - # characters. Example: stu-3

  Located in linkedList.h & linkedList.c & the appropriate .c and .h files for the type because printList is passed a function pointer.

- clearList – clears all the words and nodes in the list. Ensure you don't leak memory for the word or the node – Located in linkedList.h & linkedList.c

- menu – displays the following menu and ensures the number entered is within range – Located in myUtils.h & myUtils.c --
  1) Print the List
  2) Add First
  3) Add Last
  4) Sort the List (ascending order)
  5) Delete a Word
  6) Quit –

Other Things to Be Done

- You must create linkedlist.c, listUtils.c and the appropriate .c for the type
- I have provided linkedlist.h and listUtils.h – You CANNOT change linkedlist.h . You can't change my code in listUtils.h.
- You may only create .c files. To start with you will need words.c.
- You will create an output file that shows the run of the program using valgrind. This output file should show that no memory is being leaked.
- I have provided cscd240Lab18.c – You can't change anything in this file.

TO TURN IN

- All code to compile and run your code
- My Makefile
- Your valgrind run named cscd240Lab18val.txt
- Output run named cscd240Lab18out.txt

You will submit a zip file named your last name first letter of your first name lab18.zip. (Example steinerslab18.zip)

Give us everything required to compile, run and test your code, otherwise you will receive a 0. If you your code seg faults you will get a 0.