CS300 Exam 2 Review

1) Consider the following representation for a linked list:

```
typedef struct Node *NodePtr;
typedef struct Node
{
  char data;
  NodePtr psNext;
} Node;
```

a) Write a program segment that opens the file "text.txt" and processes the first line in the file as follows: Read each character one at a time and place each character in a Node of a singly linked list pointed to by a NodePtr psLine. For instance, if the first line in the file is AB CD, the list would be A->B-> ->C->D

b) If you were going to turn this into a function, what would the function header look like? Why?

c) Write a function that accepts the pointer psLine and prints the list forwards.

d) Write a function that accepts the pointer psLine and prints the list backwards. Do this without reversing the list.

e) Write a function that will remove (and free) every node that contains a space.

f) Write a function that will reverse the list: A->B-> ->C->D becomes D->C-> ->B->A

g) Given two linked lists, insert nodes of second list into first list at alternate positions of first list. Assume that the second list is always shorter than the first list. The second list must remain unchanged.
List 1: 1->5->7->3->9
List 2: 6->0->2->4,
New List 1: 1->6->5->0->7->2->3->4->9

You should be able to perform the above operations if the linked list is represented as follows:

```
typedef struct List *ListPtr;
typedef struct List
{
  int numElements;
  NodePtr *psHead;
} List;
```

```
typedef struct Node *NodePtr;
typedef struct Node
{
  char data;
  NodePtr psNext;
} Node;
```

h) Two ListPtrs psLine1 and psLine2 point to random lines from the data file text.txt. We would like to merge the line pointed to by psLine2 to the end of the line pointed to by psLine1. If the length of psLine1 is p characters, and the length of psLine2 is q characters, what is the computing complexity for merging the lines as discussed?

i) An ordered list is pointed to by a pointer psList. The ordering is characters from smallest ASCII value to the largest ASCII value. Can we search this list for a specific character and return a pointer to the node, if the character exists, in $O(\log_2 N)$ time? Why or why not?

2) Using the List API, how would you implement the Stack functions?

3) Make sure you understand Makefiles

4) A function has $5n^2-6n$ computations. Is the running time for this function $O(n^2)$? Why or why not?

5) Understand singly linked, singly linked circular, doubly linked, and doubly linked circular lists. You need to be able to write code for any of the list representations and understand the computational complexity regarding a specific list operation with a give list representation.

6) An ordered array is searched for a specific value. What is the worst case computing complexity for this operation? Explain your answer.

7) Review representations of strings and implementing string functions given the various representations.

8) Review representations of queues, static vs dynamic, singly linked vs singly linked circular, circular vs non-circular static queue

9) How do data structures change with the use of a union? How do unions work?

10) Using the definition of big-O notation, show that the computing complexity of $N^2 + \log_2 N + 4$ is $O(n^2)$.