Name:		**section**
(as it would appear on official course roster)		
Umail address:	**@umail.ucsb.edu**	
Optional: name you wish to be called if different from name above.		
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone"		

1

h09

CS16 F19

h09: Chapter 13 More Linked lists

ready?	assigned	due	points
true	Sat 11/16 11:00AM	Sat 11/23 11:59PM	30

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

UPLOAD A PDF OF YOUR ANSWERS TO GRADESCOPE BEFORE THE DUE DATE. There is NO MAKEUP for missed assignments;

Read Chapter 13, section 13.1 (pages 740- 759).

PLEASE MARK YOUR HOMEWORK CLEARLY, REGARDLESS OF IF YOU WRITE IT OUT IN INK OR PENCIL!

Please:

- **No Staples**.
- **No Paperclips**.
- **No folded down corners**.

1.(10 pts) Consider a linked list where each node is of the same type as in the previous question. Complete the definition of the function deleteNode given below, that takes as input a pointer to the head of the list, and an integer value. The function should delete all the nodes in the list whose data members have the given value. If the list is empty or if there is no node with the given value, the function should not do anything.

```
void deleteNode(struct Node*& head, int value);
```

For all the following questions, use the definitions of the struct Node and struct LinkedList from lab 07.

2.(10 pts) Implement a function that returns the number of even elements in a linked list. Test your code before writing it out. Illegible code will receive 0 credit. Below is the declaration of the function

```
int countEven(LinkedList* list);
```

2

h09

CS16 F19

3

h09

CS16 F19

3.(10 pts) Implement a function that takes a linked list and the number of elements in the list as input and returns a dynamic array containing the data elements of the linked list. Test your code before writing it out.

```
int* linkedListToArray(LinkedList * list, int len);
```