1) Big O(1), O(n), O(n^2), O(n^3)
    a) O(n^3) because n*n*n = n^3 and the loop goes through n*n*n cycles.
    b) O(n) because the loop goes through n cycles
    c) O(1) because this code will only loop once no matter what n is. However, it also won't compile because the curly bracket is facing the wrong way.
    d) O(n^2) because it will go through n loops and then as the first video states it will go through n(n+1)/2 loops which comes out to n^2 so O(n^2)
    e) O(n) Y is hard coded to 10 so it will loop 10 times no matter what c = 10 so this is O(n)
    f) O(n) This code goes through 2 different loops that are n times large so this 2n or in other words O(n)
2) Recursion
    a) if the next node in the linked list is null than we are at the tail of the linked list
    b) The total sum is changed as well as the current node