[**Closed book and notes.**] Assume ArrayList and LinkedList refer to the implementations developed in lecture. If asked to implement a method, you may not assume that any other methods from the class have already been implemented unless explicitly stated. Show all of your work clearly in the space provided or on the additional page at the end of the exam. If the additional page is used, clearly identify to which exam question it is related. Be sure to **read each problem carefully**. Note that the exam is double sided.

```
1  public static int countOdd(List<Integer> numbers) {
2    if(numbers==null) {
3      throw new IllegalArgumentException("List cannot be null");
4    }
5    int count = 0;
6    for(Integer number : numbers) {
7      if(number%2!=0) {
8        count++;
9
10     }
11   }
12   return count;
13 }
```

Assume n is the size of the List passed to countOdd()

1. (20 points) True/False (**T** or **F**)

_____ The asymptotic time complexity for countOdd is either $O(n)$ or $O(n^2)$ depending on whether the method is passed an ArrayList or a LinkedList.

_____ The asymptotic time complexity would be different for a list containing all even numbers than for a list containing all odd numbers.

_____ numbers could be a Collection<Integer> instead of List<Integer> and not require **any** other changes.

Suppose we pass an object from a LinkedList to countOdd(). For countOdd() to function correctly, the following methods must be implemented in the LinkedList class: (indicate T or F for each method)

_____ size()

_____ iterator()

_____ isOdd()

_____ iterable()

Suppose numbers.get(count); is inserted on line 9.

_____ countOdd() will crash.

_____ The asymptotic time complexity for countOdd() will be $O(n)$ when operating on a java.util.ArrayList.

_____ The asymptotic time complexity for countOdd() will be $O(n)$ when operating on a java.util.LinkedList.

2. (10 points) Describe the differences in the java.util.ArrayList implementation compared to the one created in lecture that affect the asymptotic time complexity of any of the methods in the List interface.

3. (10 points) Describe the limitations of Big-O notation.

4. (5 points) Suppose $T(n) = 3n^2 + 8$. Determine the Big-O notation.

5. Recall that our ArrayList had one attribute: data and our LinkedList had one attribute: head.
(a) (5 points) Implement ArrayList.size().

(b) (10 points) Implement LinkedList.size().

6. (20 points) Suppose a second attribute, tail, that points to the last element in our list is added to our LinkedList. Implement the add(E element) method for the modified class.

7.

(a) (10 points) Complete the implementation of the following method **without** using the enhanced for loop.

```
public static double findThirdLargest(Collection<Double> numbers) {
```

(b) (5 points) Give the Big-O time complexity for your implementation of in part **(a)**.

(c) (5 points) Justify your answer to part **(b)**.

Additional work area for any problem. Clearly identify which problem is associated with the work on this page.