(a) Suppose you had an application that required you to frequently add to the front of a list. Which data structure would be preferable to use: ArrayList or LinkedList? Justify your answer.

(b) Circle the bullet in front of each statement that accurately describes your instructor's course policies:

- Having a copy of all or part of another student's source code, just to look at for a little help, is considered cheating.
- All students using a laptop during class must email a copy of their lecture notes to me before the end of the day.
- Your instructor will not accept a late lab submission if you have already been late with at least two prior labs.
- The final exam for this course will be common with the other sections of CS2852.
- For each lecture period you miss, your grade will be docked by progressively larger amounts.
- A non-functional lab submitted on time will receive an automatic 50% penalty.
- Your instructor wants to eat lunch with you.

Do not circle the bullet in front of any statement that incorrectly describes your instructor's course policies.
Feel free to provide an explanation for your answer if you believe the statement to be ambiguous.

Implement the ArrayList.add(E) and ArrayList.get(int) methods using the same assumptions that we made in lecture.

(a) Implement the indexOf method for the LinkedList class we have been developing in lecture. Recall that the class has one attribute: Node head where Node is an inner class with two attributes: E value and Node next and two constructors: Node(E value) and Node(E value, Node next).

(b) Give the Big-O time complexity for the following method when a LinkedList of size n is passed as an argument. Justify your answer.

```
public static int sum(List<Integer> numbers) {
  int sum = 0;
  for(int i=0; i<numbers.size(); ++i) {
    sum += numbers.get(i);
  }
  return sum;
}
```

Write a recursive helper method for a BinaryTree<E> class that will return the height of a subtree and is compatible with the following method:

```
public int height() {
  return height(root);
}
```

Complete the implementation of the remove() method in the following class uses a hash table to implement the Set interface.

```
public class HashQuiz<E> implements Set<E> {
  private ArrayList<E>[] table;

  public HashSet() {
    table = (ArrayList<E>[])new ArrayList[7919];
  }

  // ...

  public boolean remove(Object target) {









  }
}
```

(a) Use big-O notation to describe the overall worst case time complexity for a recursive method that calculates the height of a balanced binary tree (e.g., the height(Node) method we implemented in lecture). Be sure to explain your reasoning.

(b) Use big-O notation to describe the overall worst case time complexity for a non-recursive method returns the largest element in a balanced binary search tree. Be sure to explain your reasoning.

What is a Map? How does it differ from a Set?

(a) Suppose the following Integers are added to a hash table with a capacity of 6. Illustrate the resulting data structure. Note that the hashCode() method for the Integer class just returns the value of the integer.

0, 8, 1, 5, 6, 7

(b) What is the load factor, L, for this hash table?