Com S 228 Fall 2014 Exam 2 Answer Key

	Code snippet	Output	List iterator and state or exception thrown
	``init ();
iter = aList.listIterator ();```	(none)	|A B C D
	``init ();
iter = aList.listIterator ();
iter.next ();
iter.add ("X");
iter.next ();
iter.add ("Y");
iter.previous ();
iter.remove ();
System.out.println (iter.nextIndex ());```	3	A X B|C D
1.	```init ();
iter = aList.listIterator (aList.size ());
iter.previous ();
iter.next ();
System.out.println (iter.nextIndex ());
iter.next ();```	4	NoSuchElementException
	``init ();
iter = aList.listIterator (aList.size ());
System.out.println (iter.previous ());
System.out.println (iter.next ());
System.out.println (iter.previous ());
iter.remove ();```	D D D	A B C|
	``init ();
iter = aList.listIterator (4);
System.out.println (iter.nextIndex ());
iter.add ("X");
iter.remove ();```	4	IllegalStateException

2.
```java
public static <T> boolean isSorted(List<T> l,
                                    Comparator<? super T> comp)
{
  ListIterator<T> f, s;

  if (l == null) {
    throw new NullPointerException();
  }

  f = l.listIterator();
  s = l.listIterator(1);

  while (s.hasNext()) {
    if (comp.compare(f.next(), s.next()) > 0) {
      return false;
    }
  }

  return true;
}
```

3. (a) Answer: $O(1)$

   (b) Answer: $O(n)$ when resizing is needed. $O(1)$ amortized cost.

4. (a)
```java
public void insertTail(Integer value)
{
  Node tmp;

  if (value == null) {
    throw new NullPointerException();
  }

  tmp = new Node(value);

  if (size == 0) {
    head = tmp;
    tail = tmp;
  } else {
    tail.next = tmp;
    tmp.prev = tail;
    tail = tmp;
  }
  size++;
}
```
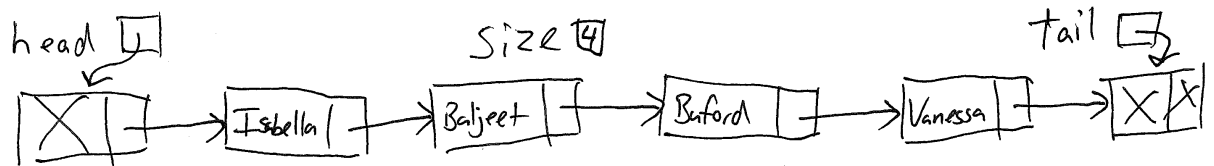
2

(b)
```
public void primeSplit(IntegerList prime,
                            IntegerList composite)
{
  Integer value;

  prime.head = null;
  prime.tail = null;
  prime.size = 0;
  composite.head = null;
  composite.tail = null;
  composite.size = 0;

  while (size != 0) {
    value = removeHead();
    if (isPrime(value)) {
      prime.insertTail(value);
    } else {
      composite.insertTail(value);
    }
  }
}
```
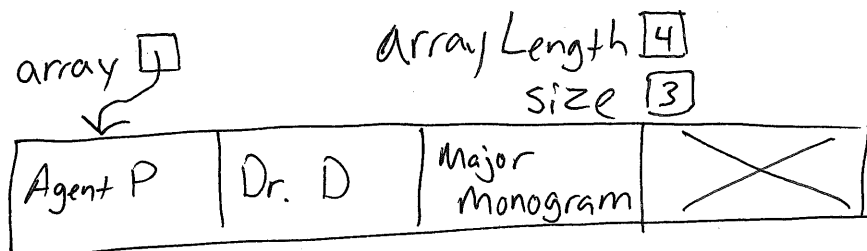
5. (a) / - ^ - ^ b 2 * 4 * a c 0.5 b * 2 a

   (b) ((b ^ 2 - 4 * a * c) ^ 0.5 - b) / (2 * a)

   (c) b 2 ^ 4 a c * * - 0.5 ^ b - 2 a * /

6. (a)

   (b)