Exercise 1 – Iterators (video §3.E)
Consider the partial code of LinkedBag.java below. The container data structure for the LinkedBag is a linked list. We need to develop a ListIterator that can traverse through the list and produce the elements in the list sequentially starting with the head of the list. (The line numbers match the downloadable version of the code on the precept page.)

A. The code for the add() method in the LinkedBag class is given below. When a new element is added to the LinkedBag, does it add the new element to the beginning or to the end of the Linked list?

```java
public class LinkedBag<Item> implements Iterable<Item> {
    private Node first;  // beginning of bag
    private int n;       // number of elements in bag

    // helper linked list class
    private class Node {
        private Item item;
        private Node next;
    }

    public void add(Item item) {
        Node oldfirst = first;
        first = new Node();
        first.item = item;
        first.next = oldfirst;
        n++;
    }
}
```

Note that all code found in the algs4 library is listed on the booksite. The url is: https://algs4.cs.princeton.edu/code/
B. Complete the missing code below starting at line 66 in iterator(), the ListIterator class declaration (line 70), ListIterator(), hasNext() and next().

```java
// Returns an instance of the iterator class (to be used for iterating)
public Iterator<Item> iterator() {
    
    // traverse through the list sequentially starting at first; use Iterator and the generic Item
    private class ListIterator implements ____________________________________ {
        private Node current;
        
        public ListIterator() {
        }

        public boolean hasNext() {
        }

        public void remove() {
            throw new UnsupportedOperationException();
        }

        public Item next() {
            if (!hasNext())
                throw new NoSuchElementException();

        }
    }
}
```

C. The code below is a client of the LinkedBag class. What output is produced by this code?

```java
LinkedBag<Integer> myBag = new LinkedBag<Integer>();
myBag.add(3);
myBag.add(1);
myBag.add(2);
for (int i : myBag) {
    for (int j : myBag) {
        StdOut.println(i + " + " + j);
    }
}
StdOut.println();
```

Exercise 2 – Elementary sorting

Give an example of the best and worst case for both insertion sort and selection sort (using an array of the integers between 0 and 4) in terms of number of compares and number of exchanges.

<table>
<thead>
<tr>
<th></th>
<th>best case</th>
<th>worst case</th>
</tr>
</thead>
<tbody>
<tr>
<td>insertion sort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>selection sort</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>