Assignment 2

Due date: October 16th, 2017
Total points: 10

1. Develop a class `ResizingArrayQueue<T> implements Iterable<T>` that implements the queue API with a resizing array. The size of the array should be kept within a constant factor of the size of the collection. In order to minimize the space and reduce array copying, you should use the array in a circular way: For example, if tail has reached the end of the array and if position 0 is not in use due to previous dequeue operations, you can add a new item at position 0 of the array.

2. Develop a `LinkedList<T> implements Iterable<T>` class with the following API:

```
public LinkedList()
public boolean isEmpty()
public int size()
public void add(T item)
public void add(T item, int index) throws IndexOutOfBoundsException
public T get(int index) throws IndexOutOfBoundsException
public T set(int index, T item) throws IndexOutOfBoundsException
public T remove(int index) throws IndexOutOfBoundsException
public Iterator<T> iterator()
```

3. Develop a `DoublyLinkedList<T> implements Iterable<T>` class with the following API:

```
public LinkedList()
public boolean isEmpty()
public int size()
public void add(T item)
public void add(T item, int index) throws IndexOutOfBoundsException
public T get(int index) throws IndexOutOfBoundsException
public T set(int index, T item) throws IndexOutOfBoundsException
public T remove(int index) throws IndexOutOfBoundsException
public Iterator<T> iterator()
public ListIterator<T> listIterator() // returns a java.util.ListIterator,
which is a subinterface of java.util.Iterator
```