The main task in this recitation is for you to fix a version of A3, doubly linked lists, so that it implements interfaces Iterator and Iterable. Do the following steps to prepare for this —you can use either your solution to A3, if you *know* it is correct, or our solution, which is available on CMS.

A. Start a new Eclipse project, call it something like r7Iterable.
B. Create a new package called LinkedList.
C. Put the correct files into the project :
    a. Download DLLTest.java from the lecture notes page and copy it into the new project. You can do this by selecting the file, doing a copy, selecting the directory src/LinkedList, and doing a paste. This file includes new test functions to test your implementation of iterable.
    b. To use OUR DLL.java: Download a3 solution from CMS and copy DLL.java into your project as above.
    c. To use YOUR DLL.java: Copy your DLL.java from your old a3 project to the new one.
D. If JUnit4 is not available in the new project, insert a new JUnit test-case class and then delete it in the usual way.
E. Look at the code at the bottom of class DLLTest. Look at how it tests interfaces Iterator and Iterable. Of course, there are errors because Iterator and Iterable have not yet been implemented by class DLL.
F. Write inner class DLLIterator with this specification and header:

```
/** An instance is an iterator over the elements of this list. */
private class DLLIterator implements Iterator
```

G. Write method iterator with this specification and header:

```
/** Return an Iterator over the elements of this list. */
public Iterator<E> iterator()
```

H. Change the class of declaration of DLL to implement Iterable<E>.
I. If you did the above correctly, both classes should have no syntax errors. And if you Run class DLLTest and made no mistakes in the above, the test should run without errors.