**Overview:** You are given `List.java`, which uses a singly-linked list to implement a deque (doubled-ended queue, pronounced “deck”) with iterators. Because it is a singly-linked list, many of the operations require $O(n)$ time. You are to modify the code (keeping the same filename) so that it is a doubly-linked list, and so that most of the operations now run in $O(1)$-time.

**Getting Started:** As in the previous homeworks, you should already have a `~/cs171/` directory. This assignment is named “hw3”, and you should copy the assignment files to `~/cs171/homeworks/hw3` just as before:

```bash
cd /home/cs171000/share/homeworks/hw3
make copy
cd ~/cs171/hw3
```

Again there is a `Makefile` defining some useful “make” commands:

```bash
make run try to compile and run your program
make turnin try to turnin your program
make help list all the commands defined by Makefile
```

There is also a driver program (`Driver.java`) which exercises a series of tests on your code; it is executed by `make run`. If it reports any errors in your code, go back and try to fix them. If you have trouble understanding the error messages from the driver, let me know.

**Requirements:** While the program contains a `previous` link, it’s not completely/correctly implemented. You should modify the program to be correctly maintaining the `previous` links in the linked list, and make it doubly-linked. Your program should run without errors under the driver. You should eliminate all the loops that were used to find a previous node (look for occurrences of the “pre” variable). This means that all operations (except `toString()`, `reverse()`, and `Iterator` construction) should run in $O(1)$ time.

We have the usual honor requirements and late-penalty, as stated on the first homework handout.

**Grading:** You must successfully “make turnin” to get a non-zero mark. You will lose points for any errors reported by the driver, and for leaving loops in operations that should take $O(1)$ time.