1.Implements a doubly linked list.

2. Each node in the linked list contains a Student object. A Student has a name, red id and a GPA

3. Add elements to a linked list. When a new Student object is added, list is maintained in lexicographical order by student’s name.

4. Given a k, returns the k'th element in the list. If k is out-of-bounds throw an exception.

5. Print out the Red Ids of the students that are on probation (GPA less than 2.85) that in the list from the front to the back of the list.

6. Print out the names of the students with GPA of 4.0 in the list from the back to the front of the list.

7. Use the strategy pattern to allow you determine how a linked list object will be ordered when a linked list object is created.

8. Implement an iterator for your linked list. Don't covert your linked list to an array or other collection to implement your iterator.

9. Use the null object pattern to represent head and tail nodes in the linked list.

10. Implement what we will for now will call OnProbationFilter.

11. Create a decorator class OnProbationDecorator that decorates your Linked list class. The decorator modifies the toString, toArray and iterator methods to return just student elements that are on probation.