CS1332 - Data Structures
Homework #1, Circular Buffer implemented with an Array

For this homework we will be creating a circular buffer. These kinds of buffers are extremely useful for buffering information coming into an application from an external source.

There are several variations of circular buffers based on their behavior -- especially when the buffer starts to fill up. One variation allows the buffer to regrow when it starts to fill so that no data is ever lost. Another variation just overwrites the data at the front (basically erasing the oldest data). The latter is often used in video streaming where we are not consuming frames fast enough. You might see the video jump a little when it "skips" some frames that have been effectively discarded by the wrapping action.

We will code a circular buffer that supports both kinds of actions. The enumeration BufferGrowMode provides the definition of the two available modes: REGROW and OVERWRITE.

You are provided with an interface for the buffer called CircularBuffer.java. **See the javadoc comments for instructions on implementing the required methods**. These comments will contain important information on how to implement the methods and required return and initialization values.Name your class CircularArrayBuffer.

When constructing a new CircularArrayBuffer, you should use a primitive array as the backing store. In class we talked about how to create an array of generic type, and how a circular array operates. You may NOT use any of the Java built-in data structures for this assignment (other than the primitive array). Your constructor should make the initial array capacity 10. Also it should set the grow mode to REGROW. *WARNING: This homework **MUST** use a circular array. Any other implementation will receive a zero.*

I am providing you a basic Intelli-J project with the required files already set up. Just use Import Project to bring in your project. If using another IDE, just ignore all the files except the 3 provided files below and bring them into the IDE of your choice. You will need to ensure Junit 4 is enabled to use the CircularBufferTest.java class.

Provided Files:
BufferGrowMode.java contains an enumeration of the options to regrow the buffer.
CircularBuffer.java contains an interface specification for the class you will create.
CircularBufferTest.java contains some JUnit code to run to test your implementation.

Files you create:
CircularArrayBuffer.java will contain your implementation of the required operations.

Submit on t-square your implementation file : CircularArrayBuffer.java.

DO NOT MAKE ANY CHANGES TO THE PROVIDED INTERFACE!!!
You may however, add any private methods and helper classes to your implementation file you desire, but you may not modify the provided interface file.

Grading Information:

1. Be sure your code compiles. Non-compiling code will be given an automatic zero.
2. Be sure to use a circular array with a backing store of a primitive array (T[]). Any other implementation will receive a zero.
3. Be sure your class is named properly so that the JUnit tests will run.
4. Be sure not to use ANY of the Java Collection classes (like ArrayList).
5. Be sure you have proper runtimes for methods. All operations must be O(1) except contains and regrow which should be O(n).
6. Assuming all the above requirements are met, and you pass all the tests in the provided JUnit, you should receive a 100% on this homework.

EXTRA CREDIT: Bug Finder. You may receive extra credit for any bugs and problems in your code that are not discovered by the provided tests. Email the instructor: watersr@cc.gatech.edu with a description of the bug and a test that would have found the bug.