

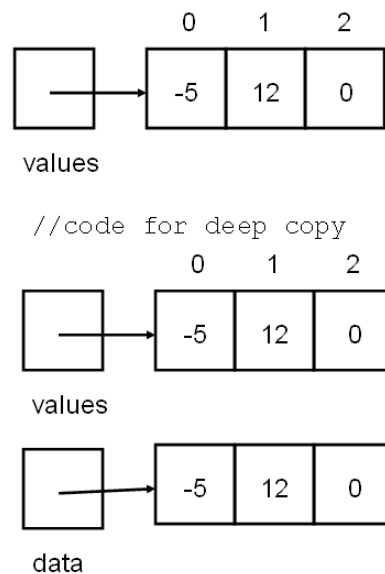
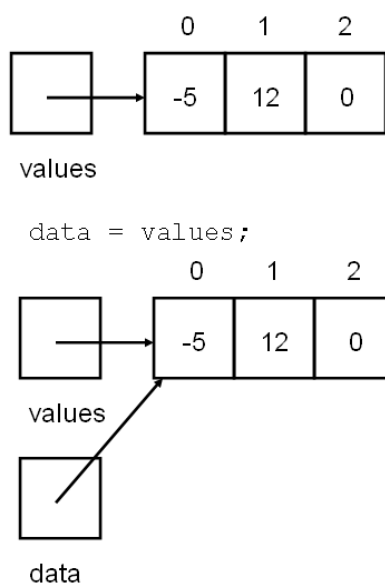
Collaboration: individual assignment. **Internet access allowed ONLY** to your text, if you have the e-version, and the Java API.

This program uses the code developed in Part 1. You can work from your solution or the solution provided by the instructor.

1. Add a **copy** method to the **Stack** class. This method returns a *deep copy* of a stack. Recall that a copy is *deep* if it contains exactly the same data as the original, but the copy and original share no memory. See the pictures below (extracted from <http://www.cs.utexas.edu/~scottm/cs307/handouts/deepCopying.htm>).

Picture on left is of a shallow copy, also an alias.

Picture on the right is of a deep copy.



Signature of your copy method: **public Stack copy()**

To implement this method, you may use only stack data structures. Create as many as you need.

After a call to the copy method, the *caller* should have two identical stacks.

2. Add a **copy** method to the **Queue** class. This method returns a *deep copy* of a queue. Recall that a copy is *deep* if it contains exactly the same data as the original, but the copy and original share no memory. (See pictures above.)

Signature of your copy method: **public Queue copy()**

To implement this method, you may use only queue data structures. Create as many as you need.

After a call to the copy method, the *caller* should have two identical queues.

3. Add calls to the main methods in both Queue and Stack classes to demonstrate that your copy methods work properly.

Create a folder named by your last name, put your files in that folder (partial submissions are better than no submission), zip/compress that folder, and upload to OAKS before the deadline.