TESTING DOUBLY LINKED LIST: LAB 3

Doubly Linked List: Stacks

```
Pushing elements onto the stack s1 & s2. . .
Is s empty?  false
Size of s:  7

Is q2 empty?  false
Size of q2:  7

Pushing elements onto s . . .

Linked list of s is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Carrot -> Banana -> Apple
-> A1 -> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s at position 5

Linked list of s is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1
-> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s at position 0

Linked list of s is:
Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 ->
C3 -> D4 -> E5 -> F6 -> G7 -> End of list

Just pop'd: Gamer from s

Linked list of s is:
French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 -> C3 -> D4
-> E5 -> F6 -> G7 -> End of list
Size of s: 12
What's the first element?  French Fry
What's at element 3?  Banana
What's at element 9?  E5

Pushing elements onto s2 . . .

Linked list of s2 is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Carrot -> Banana -> Apple
-> A1 -> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s2 at position 5

Linked list of s2 is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1
-> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s2 at position 0
```

```
Linked list of s2 is:
Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 ->
C3 -> D4 -> E5 -> F6 -> G7 -> End of list

Linked list of s is:
French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 -> C3 -> D4
-> E5 -> F6 -> G7 -> End of list

Is s == s2?  false


Linked list of s2 is:
Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 ->
C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Size of s2:  13
What's the first element?  Gamer
What's at element 6?  A1
What's at element -2?  null

Just pop'd:  Gamer from s2

Linked list of s2 is:
French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 -> C3 -> D4
-> E5 -> F6 -> G7 -> End of list

Linked list of s is:
French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 -> C3 -> D4
-> E5 -> F6 -> G7 -> End of list

Is s2 == s? true

Just empty'd s!
Is s empty?  true
What's the size of s?  0
s at index 0:  null
s at index 1:  null
s at index 2:  null

Is s2 empty?  false
What's the size of s2?  12
s2 at index 0:  French Fry
s2 at index 1:  Eggs
s2 at index 2:  Dumpling
s2 at index 3:  Banana
s2 at index 4:  Apple
s2 at index 5:  A1
s2 at index 6:  B2
s2 at index 7:  C3
s2 at index 8:  D4
s2 at index 9:  E5
s2 at index 10:  F6
s2 at index 11:  G7
s2 at index 12:  null
```

```
s2 at index 13:  null
s2 at index 14:  null
s2 at index -2:  null

Just empty'd s2!
Is s2 empty?  true
What's the size of s2?  0
s2 at index 0:  null
s2 at index 1:  null
s2 at index 2:  null
```

Doubly Linked List: Queues

```
Is q empty?  true
Size of q:  0

Is q2 empty?  true
Size of q2:  0

Enqueue'ing elements into q . . .

Linked list of q is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
Heap -> End of list
Removing element in q at position 7

Linked list of q is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
End of list
Removing element in q at position 0

Linked list of q is:
Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Just dequeue'd: Banana from q

Linked list of q is:
Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list
Size of q: 5
What's the first element?  Carrot
What's the last element?  Gamer
What's at element 3?  French Fry
What's at element 9?  null

Enqueue'ing elements onto q2 . . .

Linked list of q2 is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
Heap -> End of list
Removing element in q2 at position 7
```

```
Linked list of q2 is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
End of list
Removing element in q2 at position 0

Linked list of q2 is:
Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Linked list of q is:
Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Is q == q2?  false


Linked list of q2 is:
Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list
Size of q2:  6
What's the first element?  Banana
What's the last element?  Gamer
What's at element 4?  French Fry
What's at element -2?  null

Just dequeue'd:  Banana from q2

Linked list of q2 is:
Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Linked list of q is:
Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Is q2 == q? true

Just empty'd q!
Is q empty?  true
What's the size of q?  0
q at index 0:  null
q at index 1:  null
q at index 2:  null

Is q2 empty?  false
What's the size of q2?  5
q2 at index 0:  Carrot
q2 at index 1:  Dumpling
q2 at index 2:  Eggs
q2 at index 3:  French Fry
q2 at index 4:  Gamer
q2 at index 5:  null
q2 at index 6:  null
q2 at index 7:  null
q2 at index 8:  null
q2 at index 9:  null
q2 at index 10:  null
q2 at index -2:  null
```

```
Just empty'd q2!
Is q2 empty?  true
What's the size of q2?  0
q2 at index 0:  null
q2 at index 1:  null
q2 at index 2:  null
```

TESTING PERSISTENT STRUCTURE: LAB 4

Persistent Structure: Stacks

```
Testing immutability. . .

Pushed 6 onto the stack?
A is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
B is:  6 -> 1 -> 2 -> 3 -> 4 -> 5 -> End of list

Popped something off the stack?
A is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
B is:  2 -> 3 -> 4 -> 5 -> End of list

Removed at index 1 from the stack?
A is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
B is:  1 -> 3 -> 4 -> 5 -> End of list

Emptied the stack?
A is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
B is:  End of list


Pushing elements onto the stack s1 & s2. . .
Is s empty?  false
Size of s:  7

Is q2 empty?  false
Size of q2:  7

Pushing elements onto s . . .

Linked list of s is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Carrot -> Banana -> Apple
-> A1 -> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s at position 5

Linked list of s is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1
```

```
-> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s at position 0

Linked list of s is:
Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 ->
C3 -> D4 -> E5 -> F6 -> G7 -> End of list

Just pop'd: Gamer from s

Linked list of s is:  French Fry -> Eggs -> Dumpling -> Banana -> Apple
-> A1 -> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Size of s: 12
What's the first element?  French Fry
What's at element 3?  Banana
What's at element 9?  E5

Pushing elements onto s2 . . .

Linked list of s2 is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Carrot -> Banana -> Apple
-> A1 -> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s2 at position 5

Linked list of s2 is:
Heap -> Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1
-> B2 -> C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Removing element in s2 at position 0

Linked list of s2 is:
Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 ->
C3 -> D4 -> E5 -> F6 -> G7 -> End of list

Linked list of s is:
French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 -> C3 -> D4
-> E5 -> F6 -> G7 -> End of list

Is s == s2?  false


Linked list of s2 is:
Gamer -> French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 ->
C3 -> D4 -> E5 -> F6 -> G7 -> End of list
Size of s2:  13
What's the first element?  Gamer
What's at element 6?  A1
What's at element -2?  null

Just pop'd:  Gamer from s2

Linked list of s2 is:
French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 -> C3 -> D4
-> E5 -> F6 -> G7 -> End of list
```

```
Linked list of s is:
French Fry -> Eggs -> Dumpling -> Banana -> Apple -> A1 -> B2 -> C3 -> D4
-> E5 -> F6 -> G7 -> End of list

Is s2 == s? true

Just empty'd s!
Is s empty?  true
What's the size of s?  0
s at index 0:  null
s at index 1:  null
s at index 2:  null

Is s2 empty?  false
What's the size of s2?  12
s2 at index 0:  French Fry
s2 at index 1:  Eggs
s2 at index 2:  Dumpling
s2 at index 3:  Banana
s2 at index 4:  Apple
s2 at index 5:  A1
s2 at index 6:  B2
s2 at index 7:  C3
s2 at index 8:  D4
s2 at index 9:  E5
s2 at index 10:  F6
s2 at index 11:  G7
s2 at index 12:  null
s2 at index 13:  null
s2 at index 14:  null
s2 at index -2:  null

Just empty'd s2!
Is s2 empty?  true
What's the size of s2?  0
s2 at index 0:  null
s2 at index 1:  null
s2 at index 2:  null
```

Persistent Structure: Queues

```
Testing immutability. . .

Enqueue'd 6 onto the queue?
C is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
D is:  1 -> 2 -> 3 -> 4 -> 5 -> 6 -> End of list

Dequeue'd something off the queue?
C is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
D is:  2 -> 3 -> 4 -> 5 -> End of list

Removed at index 1 from the queue?
C is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
```

```
D is:  1 -> 3 -> 4 -> 5 -> End of list

Emptied the queue?
C is:  1 -> 2 -> 3 -> 4 -> 5 -> End of list
D is:  End of list


Is q empty?  true
Size of q:  0

Is q2 empty?  true
Size of q2:  0

Enqueue'ing elements into q . . .

Linked list of q is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
Heap -> End of list
Removing element in q at position 7

Linked list of q is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
End of list
Removing element in q at position 0

Linked list of q is:
Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Just dequeue'd: Banana from q

Linked list of q is:  Carrot -> Dumpling -> Eggs -> French Fry -> Gamer
-> End of list
Size of q: 5
What's the first element?  Carrot
What's the last element?  Gamer
What's at element 3?  French Fry
What's at element 9?  null

Enqueue'ing elements onto q2 . . .

Linked list of q2 is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
Heap -> End of list
Removing element in q2 at position 7

Linked list of q2 is:
Apple -> Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer ->
End of list
Removing element in q2 at position 0

Linked list of q2 is:
Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list
```

```
Linked list of q is:
Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Is q == q2?  false


Linked list of q2 is:
Banana -> Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list
Size of q2:  6
What's the first element?  Banana
What's the last element?  Gamer
What's at element 2?  Dumpling
What's at element -2?  null

Just dequeue'd:  Banana from q2

Linked list of q2 is:
Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Linked list of q is:
Carrot -> Dumpling -> Eggs -> French Fry -> Gamer -> End of list

Is q2 == q? true

Just empty'd q!
Is q empty?  true
What's the size of q?  0
q at index 0:  null
q at index 1:  null
q at index 2:  null

Is q2 empty?  false
What's the size of q2?  5
q2 at index 0:  Carrot
q2 at index 1:  Dumpling
q2 at index 2:  Eggs
q2 at index 3:  French Fry
q2 at index 4:  Gamer
q2 at index 5:  null
q2 at index 6:  null
q2 at index 7:  null
q2 at index 8:  null
q2 at index 9:  null
q2 at index 10:  null
q2 at index -2:  null

Just empty'd q2!
Is q2 empty?  true
What's the size of q2?  0
q2 at index 0:  null
q2 at index 1:  null
q2 at index 2:  null
```