4.1.1

Draw the queue in Figure 4.3 as it will appear after the insertion of customer Harris and the removal of one customer from the queue. Which customer is removed? How many customers are left?

Original Queue (Fig 4.3)

Thome
Abreu
Jones

After insertion of Harris

Thome
Abreu
Jones
Harris

After removal of customer Thome.

Abreu
Jones
Harris

4.2.1

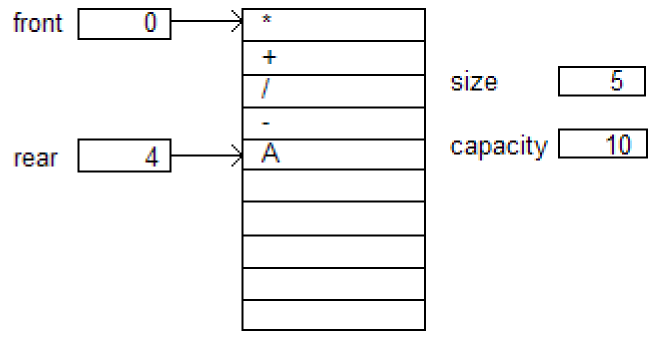
Write an algorithm to display all the elements in a queue using just the queue operations. How would your algorithm change the queue?

```
declare an integer x
declare an integer y
declare an object temp

set x = queue.getSize()

set y = 0

loop while y < x
        set temp = queue.remove()
        display temp
        queue.add(temp)
        increment y by 1
next loop
```

This algorithm would rotate the entire queue once, but the queue would essentially be the same in the end, with the same size and order of objects.

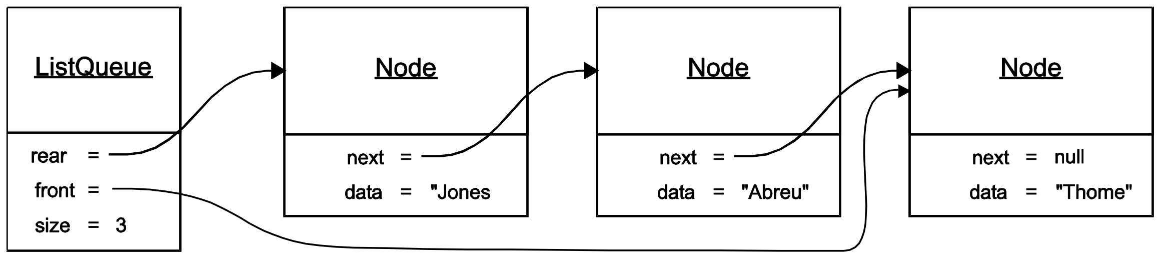
4.3.1

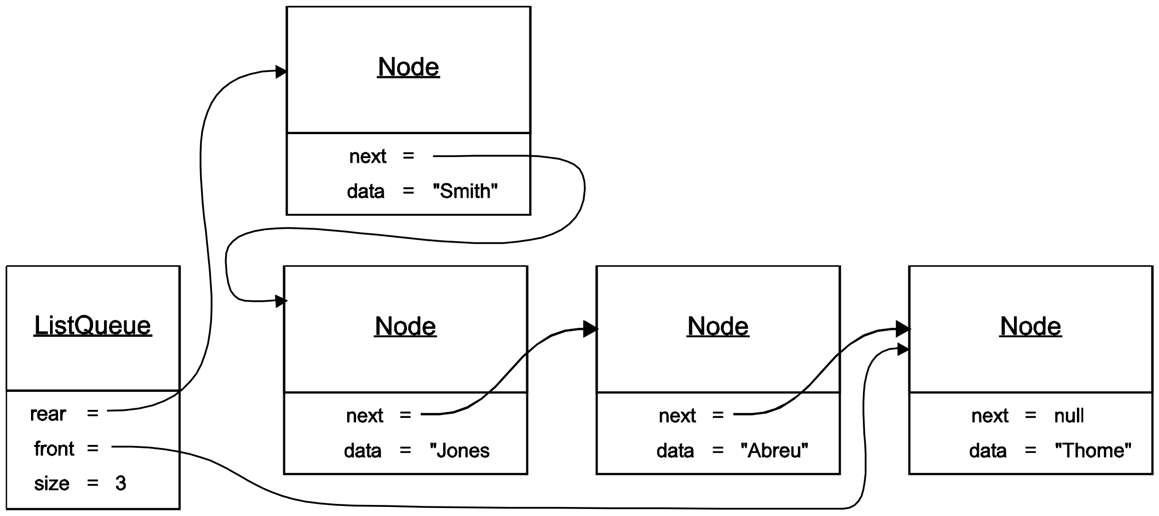
Show the new array for the queue in Figure 4.10 after the array size is doubled.

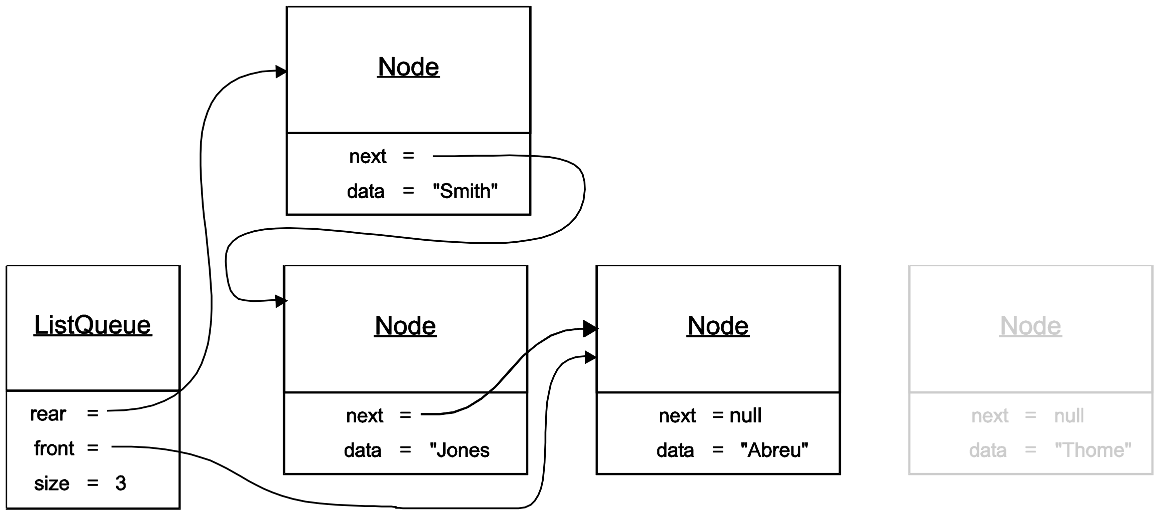


4.3.3

Redraw the queue in Figure 4.7 so that `rear` references the list head and `front` references the list tail. Show the queue after an element is inserted and an element is removed. Explain why the approach used in the book is better.







While inserting a new item at the rear is easy, to remove an item from the front you need to walk along the linked list until you reach the next to last node, so that you have a reference to the new front, this is an O(n) operation. By having the front at the head of the linked list and the rear at the tail, both insertion and removal are constant time operations.

4.4.1

For object `stackOfStrings` declared above, replace each stack operation with the appropriate `Deque` method and explain the effect of each statement in the following fragment.

```
stackOfStrings.push("Hello");
String one = stackOfStrings.pop();
if (!stackOfStrings.isEmpty())
    System.out.println(stackOfStrings.peek());
stackOfStrings.push("Good bye");
for (String two : stackOfStrings)
    System.out.println(two);
```

Re-written using deque operations

```
stackOfStrings.addFirst("Hello");                   // Makes "Hello" the first item on the stack
String one = stackOfStrings.removeFirst();          // Removes "Hello" and sets one to refer to it.
if (!stackOfStrings.isEmpty())                      // Tests to see if the stack is empty
    System.out.println(stackOfStrings.peekFirst()); // Does not execute since stack is empty
stackOfStrings.addFirst("Good bye");                // Makes "Good bye" the first item on the stack
for (String two : stackOfStrings)                   // Starts a loop through the stack contents
    System.out.println(two);                        // Prints "Good bye"
```

4.4.3

Would the following statements execute without error? If your answer is “yes,” what would their effect be? If “no,” why not?

```
        stackOfStrings.offer("away");
        String three = stackOfStrings.remove();
```

These statement would execute without error. After offer("away") the stack would contain:

Good bye
away

After the remove, the stack would conatin:

away

4.5.1

Show the output that would be generated by running the simulation program for 20 minutes with the following passenger arrivals when `showAll` is true and `frequentFlyerMax` is 1.

A frequent flyer passenger arrives at clock = 0 and service time is 2
A frequent flyer passenger arrives at clock = 1 and service time is 1
A frequent flyer passenger arrives at clock = 3 and service time is 3
A frequent flyer passenger arrives at clock = 5 and service time is 2
A regular passenger arrives at clock = 0 and service time is 1
A regular passenger arrives at clock = 1 and service time is 1
A regular passenger arrives at clock = 2 and service time is 1

A regular passenger arrives at clock = 3 and service time is 1
A regular passenger arrives at clock = 4 and service time is 2

```
Time is 0: Frequent Flyer arrival, new queue size is 1
Time is 0: Regular Passengers arrival, new queue size is 1
Time is 0: Serving Frequent Flyer with time stamp 0
Time is 1: Frequent Flyer arrival, new queue size is 1
Time is 1: Regular Passengers arrival, new queue size is 2
Time is 2: Regular Passengers arrival, new queue size is 3
Time is 2: Serving Frequent Flyer with time stamp 1
Time is 3: Frequent Flyer arrival, new queue size is 1
Time is 3: Regular Passengers arrival, new queue size is 4
Time is 3: Serving Regular Passengers with time stamp 0
Time is 4: Regular Passengers arrival, new queue size is 4
Time is 4: Serving Frequent Flyer with time stamp 3
Time is 5: Frequent Flyer arrival, new queue size is 1
Time is 7: Serving Frequent Flyer with time stamp 5
Time is 9: Serving Regular Passengers with time stamp 1
Time is 10: Serving Regular Passengers with time stamp 2
Time is 11: Serving Regular Passengers with time stamp 3
Time is 12: Serving Regular Passengers with time stamp 4
Time is 14:  Server is idle
Time is 15:  Server is idle
Time is 16:  Server is idle
Time is 17:  Server is idle
Time is 18:  Server is idle
Time is 19:  Server is idle
```

4.5.3

Answer Self-Check Exercise 1 when `frequentFlyerMax` is 3.

```
Time is 0: Frequent Flyer arrival, new queue size is 1
Time is 0: Regular Passengers arrival, new queue size is 1
Time is 0: Serving Frequent Flyer with time stamp 0
Time is 1: Frequent Flyer arrival, new queue size is 1
Time is 1: Regular Passengers arrival, new queue size is 2
Time is 2: Regular Passengers arrival, new queue size is 3
Time is 2: Serving Frequent Flyer with time stamp 1
Time is 3: Frequent Flyer arrival, new queue size is 1
Time is 3: Regular Passengers arrival, new queue size is 4
Time is 3: Serving Frequent Flyer with time stamp 3
Time is 4: Regular Passengers arrival, new queue size is 5
Time is 5: Frequent Flyer arrival, new queue size is 1
Time is 6: Serving Frequent Flyer with time stamp 5
Time is 8: Serving Regular Passengers with time stamp 0
Time is 9: Serving Regular Passengers with time stamp 1
Time is 10: Serving Regular Passengers with time stamp 2
Time is 11: Serving Regular Passengers with time stamp 3
Time is 12: Serving Regular Passengers with time stamp 4
Time is 14:  Server is idle
Time is 15:  Server is idle
Time is 16:  Server is idle
Time is 17:  Server is idle
Time is 18:  Server is idle
Time is 19:  Server is idle
```