

// the item cannot be added to the deque, true otherwise.

```
public Object removeFirst();
// Delete and return the first item in the deque if the deque
// is not empty, otherwise return null (the deque was empty).

public Object removeLast();
// Delete and return the last item in the deque if the deque
// is not empty, otherwise return null (the deque was empty).

public Object peekFirst();
// Return the first item in the deque if the deque is
// not empty, leaving the deque unchanged. Otherwise return
// null (the deque was empty).

public Object peekLast();
// Return the last item in the deque if the deque is
// not empty, leaving the deque unchanged. Otherwise return
// null (the deque was empty).
} // end Deque
```

- a. Create a reference-based implementation of the *Deque* interface.
  - b. Create an array-based implementation of the *Deque* interface.
8. Implement the event-driven simulation of a bank that this chapter described. A queue of arrival events will represent the line of customers in the bank. Maintain the arrival events and departure events in an ADT event list, sorted by the time of the event. Use a reference-based implementation for the ADT event list.

The input is a text file of arrival and transaction times. Each line of the file contains the arrival time and required transaction time for a customer. The arrival times are ordered by increasing time.

Your program must count customers and keep track of their cumulative waiting time. These statistics are sufficient to compute the average waiting time after the last event has been processed.

Display a trace of the events executed and a summary of the computed statistics (total number of arrivals and average time spent waiting in line). For example, the input file shown in the left columns of the following table should produce the output shown in the right column.

Input File	Output
------------	--------

1	5	Simulation Begins
2	5	Processing an arrival event at time: 1
4	5	Processing an arrival event at time: 2
20	5	Processing an arrival event at time: 4
22	5	Processing a departure event at time: 6
24	5	Processing a departure event at time: 11
26	5	Processing a departure event at time: 16
28	5	Processing an arrival event at time: 20
30	5	Processing an arrival event at time: 22
88	3	Processing an arrival event at time: 24
		Processing a departure event at time: 25