## Homework

1. Augment the ArrayQueue implementation with a new rotate( ) method having semantics identical to the combination, enqueue(dequeue( )). But, your implementation should be more efficient than making two separate calls (for example, because there is no need to modify the size).

public void rotate() {

if (size > 0) {

E first = data[front];

dequeue();

enqueue(first);

}

}

1. Implement the clone( ) method for the ArrayQueue class.  
   public ArrayQueue<E> clone() {

try {

ArrayQueue<E> cloned = (ArrayQueue<E>) super.clone();

cloned.data = data.clone();

return cloned;

} catch (CloneNotSupportedException e) {

throw new InternalError(e.toString());

}

}

1. Implement a method with signature concatenate(LinkedQueue Q2) for the LinkedQueue class that takes all elements of Q2 and appends them to the end of the original queue. The operation should run in O(1) time and should result in Q2 being an empty queue.

public void concatenate(LinkedQueue<E> Q2) {

if (!Q2.isEmpty()) {

if (isEmpty()) {

front = Q2.front;

rear = Q2.rear;

size = Q2.size;

} else {

rear.next = Q2.front;

rear = Q2.rear;

size += Q2.size;

}

Q2.front = null;

Q2.rear = null;

Q2.size = 0;

}

}

1. Use a queue to solve the Josephus Problem.

public static int josephusProblem(int n, int k) {

ArrayQueue<Integer> queue = new ArrayQueue<>();

for (int i = 1; i <= n; i++) {

queue.enqueue(i);

}

while (queue.size() > 1) {

for (int i = 0; i < k - 1; i++) {

queue.enqueue(queue.dequeue());

}

queue.dequeue();

}

return queue.dequeue();

}

1. Use a queue to simulate Round Robin Scheduling.

public static void roundRobinScheduling(ArrayQueue<Process> processes, int timeQuantum) {

while (!processes.isEmpty()) {

Process currentProcess = processes.dequeue();

if (currentProcess.getTimeRemaining() > timeQuantum) {

currentProcess.execute(timeQuantum);

processes.enqueue(currentProcess); // Add back to end of queue

} else {

currentProcess.execute(currentProcess.getTimeRemaining());

System.out.println("Process " + currentProcess.getId() + " completed");

}

}

}