**ArrayDeque, PriorityQueue**

1. **ArrayDeque** in Java provides a way to apply resizable-**array** in addition to the implementation of the Deque interface. It is also known as Array Double Ended Queue or Array Deck. This is a special kind of array that grows and allows users to add or remove an element from both the sides of the queue. When implementing ArrayDeque remember to use array(not list structure) E[ ].

**Implement** ArrayDeque class that has:

* ArrayDeque() – constructor //creates array of size 16 cells
* ArrayDeque(int) – constructor //creates array of given size
* int size() – returns size of inserted objects
* boolean isEmpty() – checks either array is empty
* E first() – returns first object(head)
* E last() – returns last object(tail)
* void doubleCapacity() – doubles the capacity of array
* void addFirst() – adds object to the head
* void addLast() – adds object to the tail
* E removeFirst() – removes object from the head by returning it
* E removeLast() – removes object form the tail by returning it

1. **PriorityQueue** is an extension of [queue](http://quiz.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/)with following properties. 1) Every item has a priority associated with it. 2) An element with high priority is dequeued before an element with low priority. 3) If two elements have the same priority, they are served according to their order in the queue.

**Implement** PriorityQueue class(also using **array** E[]) that has:

* PriorityQueue(int) - creates array of given size
* void insert(E) – inserts the element
* E remove() – removes element with highest priority
* E peek() – returns element with highest priority
* boolean isEmpty() – checks either array is empty

\*The same classes could be implemented using lists. But LinkedList and DoubleLinkedLists were your previous exercises so in this tasks you should implement using simple arrays.