Design your implementation of the circular double-ended queue (deque).

Your implementation should support following operations:

* MyCircularDeque(k): Constructor, set the size of the deque to be k.
* insertFront(): Adds an item at the front of Deque. Return true if the operation is successful.
* insertLast(): Adds an item at the rear of Deque. Return true if the operation is successful.
* deleteFront(): Deletes an item from the front of Deque. Return true if the operation is successful.
* deleteLast(): Deletes an item from the rear of Deque. Return true if the operation is successful.
* getFront(): Gets the front item from the Deque. If the deque is empty, return -1.
* getRear(): Gets the last item from Deque. If the deque is empty, return -1.
* isEmpty(): Checks whether Deque is empty or not.
* isFull(): Checks whether Deque is full or not.

**Example:**

MyCircularDeque circularDeque = new MycircularDeque(3); // set the size to be 3

circularDeque.insertLast(1); // return true

circularDeque.insertLast(2); // return true

circularDeque.insertFront(3); // return true

circularDeque.insertFront(4); // return false, the queue is full

circularDeque.getRear(); // return 2

circularDeque.isFull(); // return true

circularDeque.deleteLast(); // return true

circularDeque.insertFront(4); // return true

circularDeque.getFront(); // return 4

**Note:**

* All values will be in the range of [0, 1000].
* The number of operations will be in the range of [1, 1000].
* Please do not use the built-in Deque library.