Deque

1. ***Deque:*** Implement a double-ended queue or deque (**pronounced “deck”**) is like a stack or a queue but supports adding and removing items at both ends. A deque stores a collection of items and supports the following API:
   * + public class **Deque**<Item> implements **Iterable**<Item>
     + **Deque**() create an empty deque
     + boolean **isEmpty**() is the deque empty?
     + int **size**() number of items in the deque
     + void **pushLeft**(Item item) add an item to the left end
     + void **pushRight**(Item item) add an item to the right end
     + Item **popLeft**() remove an item from the left end
     + Item **popRight**() remove an item from the right end

**API** for a generic double-ended queue

**Input Format:**

* The first line of the input contains the number of operations in total to be performed on the Deque.
* From the second line onwards each input contains an operation that should be performed on Deque. Check for the sample input.

**Output Format:**

* After each input operation, print the values of the Deque. Check for the sample output.

**Note:** Print the values of the Deque when pushLeft, pushRight, popLeft and popRight.

**Note:** Print the total number of elements in the Deque for the size operation.

**Note:** Print true if the Deque is empty for isEmpty operation, otherwise false.

**Sample Input #1:**

**11**

**isEmpty**

**size**

**pushLeft 10**

**pushRight 20**

**pushRight 30**

**pushLeft 40**

**size**

**isEmpty**

**popLeft**

**popRight**

**size**

**Sample Output #1:**

**true**

**0**

**[10]  
[10, 20]**

**[10, 20, 30]**

**[40, 10, 20, 30]**

**4**

**false**

**[10, 20, 30]**

**[10, 20]**

**2**