LinkedList, DoubleLinkedList (10 days)

**Task 1:** Implement LinkedList generic class so that it has methods:

- add(E e) //Appends the specified element to the end of this list.

- add(int index, E element) //Inserts the specified element at the specified position in this list.

- addAll(LinkedList<T> list2) //Appends all of the elements in the specified list2 to the end of this list.

- addFirst(E e) //Inserts the specified element at the beginning of this list.

- clear() //Removes all of the elements from this list.

- clone() //Returns a shallow copy of this LinkedList.

- get(int index) //Returns the element at the specified position in this list.

- getFirst() //Returns the first element in this list.

- getLast() //Returns the last element in this list.

- indexOf(Object o) //Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.

- remove() //Retrieves and removes the head (first element) of this list.

- remove(int index) //Removes the element at the specified position in this list.

- remove(E o) //Removes the first occurrence of the specified element from this list, if it is present.

- set(int index, E element) //Replaces the element at the specified position in this list with the specified element.

- size() //Returns the number of elements in this list.

- print() //Prints all values from beginning till end.

**Task 2:** Implement DoubleLinkedList generic class so that it has methods:

- add(E e, boolean head) //Appends the specified element to the end of this list(if add(obj, false)). Otherwise adds the object to the head.

- add(int index, E element) //Inserts the specified element at the specified position in this list.

(If index is closer to the tail, so iterate from end of a list)

- get(int index) //Returns the element at the specified position in this list.

(If index is closer to the tail, so iterate from end of a list)

- getFirst() //Returns the first element in this list.

- getLast() //Returns the last element in this list.

- remove(boolean head) //Retrieves and removes the head (if remove(true)) of this list.

If remove(false) - remove the last one.

- remove(int index) //Removes the element at the specified position in this list.

(If index is closer to the tail, so iterate from end of a list)

- set(int index, E element) //Replaces the element at the specified position in this list with the specified element. (If index is closer to the tail, so iterate from end of a list)

- size() //Returns the number of elements in this list.

- print(boolean head) //Prints all values from beginning till end(if print(true), otherwise from end till beginning).

**Task 3:** Implement one more LinkedList<E> class, where E is comparable object. This class must have following methods: pl

- addSorted(E obj) //Adds obj to its place(compares all object and finds where toace obj)

- remove(int index) //Removes the element at the specified position in this list.

- print() //Prints all values from beginning till end.