```
public class CAList <E> implements List<E> {
    E[] contents;
    int size;
    int start;

    @SuppressWarnings("unchecked")
    public CAList (int capacity) {
        this.contents = (E[])(new Object[capacity]);
        this.size = 0;
        this.start = 0;
    }

    private int indexFor (int index) {
        int ans = ( this.start + index) % this.contents.length ;
        System.out.println ("Index for " + index + " is " + ans );
        return ans ;
    }

    @SuppressWarnings("unchecked")
    private void expandCapacity () {
        int currentCapacity = this.contents.length ;
        if( this.size < currentCapacity ) { return; }









    }

    public E get(int index) {
        // ASSUME index is in bounds
        int toLookup = this.indexFor (index);
        return this.contents [ toLookup ];
    }

    public void prepend(E value) {
        expandCapacity ();
        this.size += 1;
        this.start - = 1;
        if ( this.start == - 1) {
            this.start = this.contents.length - 1;
        }
        this.contents [ this.start ] = value;
    }

    public void add(E value) {
        expandCapacity ();
        this.contents [ this.indexFor ( this.size )] = value;
        this.size += 1;
    }

    public String toString () {
        return java.util.Arrays.deepToString ( this.contents );
    }

    public int size() {
        return this.size ;
    }

    public static void main(String[] args ) {
        CAList <Integer> a = new CAList <>(30);
        System.out.println (a);
        a.prepend (30);
        System.out.println (a);
        a.add (40);
        System.out.println (a);
        a.prepend (20);
        System.out.println (a);
        a.add (70);
        System.out.println (a);
    }
}
```

	get(index)	add(val)	prepend(val)	remove(val)
AList				
CAList (front of sheet)				
LList				

```
class Node<T> {
    Node<T> next;
    Node<T> prev;
    T value;
}
```