ICS-202 Lab-03 Report

Mohammed Busaleh - 202158210

Exercise 2:

Code:

```java
public void printReverse() {
    DLLNode<T> t = tail;
    System.out.print(t.info + "   ");
    while (t.prev != null) {
        t = t.prev;
        System.out.print(t.info + "   ");
    }
    System.out.println(" ");
}
```

Output:

```
original: a0 a1 a2 a3 a4
reversed: a4   a3   a2   a1   a0
```

Exercise 3:

Code:

```
public void delete7() {
    int counter = 1;
    DLLNode<T> node = head;

    if (node.next == null) { //only one element is there
        head = tail = null;
        return;
    }

    while (counter < 7) {
        while (node.next != null && counter < 7) {
            node = node.next;
            counter++;
        }
        while (node.prev != null && counter < 7) {
            node = node.prev;
            counter++;
        }
    }


    if (node.next == null) {
        tail = node.prev;
        tail.next = null;
    } else if (node.prev == null) {
        head = node.next;
        head.prev = null;
    } else {
        node.prev.next = node.next;
        node.next.prev = node.prev;
    }
}
```

```
DLL<Integer> mydll = new DLL<>();
for(int i = 0; i<10; i++)
    mydll.addToTail((int)(10*Math.random()));

System.out.print("\n\nInitial list: ");
mydll.printAll();

mydll.delete7();
System.out.print("\nThe list after deleting 7th element: ");
mydll.printAll();

while(!mydll.isEmpty()) {
    mydll.delete7();
    System.out.print("\nThe list after deleting 7th element again: ");
    mydll.printAll();
}
```

Output:

```
Initial list: 0 4 1 5 9 1 2 1 2 1

The list after deleting 7th element: 0 4 1 5 9 1 1 2 1

The list after deleting 7th element again: 0 4 1 5 9 1 2 1

The list after deleting 7th element again: 0 4 1 5 9 1 1

The list after deleting 7th element again: 0 4 1 5 9 1

The list after deleting 7th element again: 0 4 1 5 1

The list after deleting 7th element again: 0 4 5 1

The list after deleting 7th element again: 4 5 1

The list after deleting 7th element again: 4 5

The list after deleting 7th element again: 5

The list after deleting 7th element again:
```

Exercise 4:

Code:

```
public void insertAlternate(DLL<T> newList) throws Exception{
    if(this.isEmpty() || newList.isEmpty()){throw new Exception("Null List/s");}
    int size1 = 1, size2 = 1;
    DLLNode<T> temp1 = this.head, temp2 = newList.head;
    while(temp1.next != null){
        temp1 = temp1.next;
        size1++;
    }
    while(temp2.next != null){
        temp2 = temp2.next;
        size2++;
    }
    if(size1 != size2){throw new Exception("Lists sizes don't match");}

    temp1 = this.head; temp2 = newList.head; //Those will be used to iterate over the lists
    DLL<T> alterList = new DLL<T>();
    do{
        alterList.addToTail(temp1.info);
        alterList.addToTail(temp2.info);
        temp1 = temp1.next;
        temp2 = temp2.next;
    }while(temp2 != null);

    this.setToNull();
    DLLNode<T> newEl = alterList.head;

    do{
        this.addToTail(newEl.info);
        newEl = newEl.next;
    }while(newEl != null);
}
```

Output:

```
Alternative list test (random with a list of zeros):
9 0 10 0 10 0 10 0 3 0 9 0 5 0 7 0 4 0 6 0
```