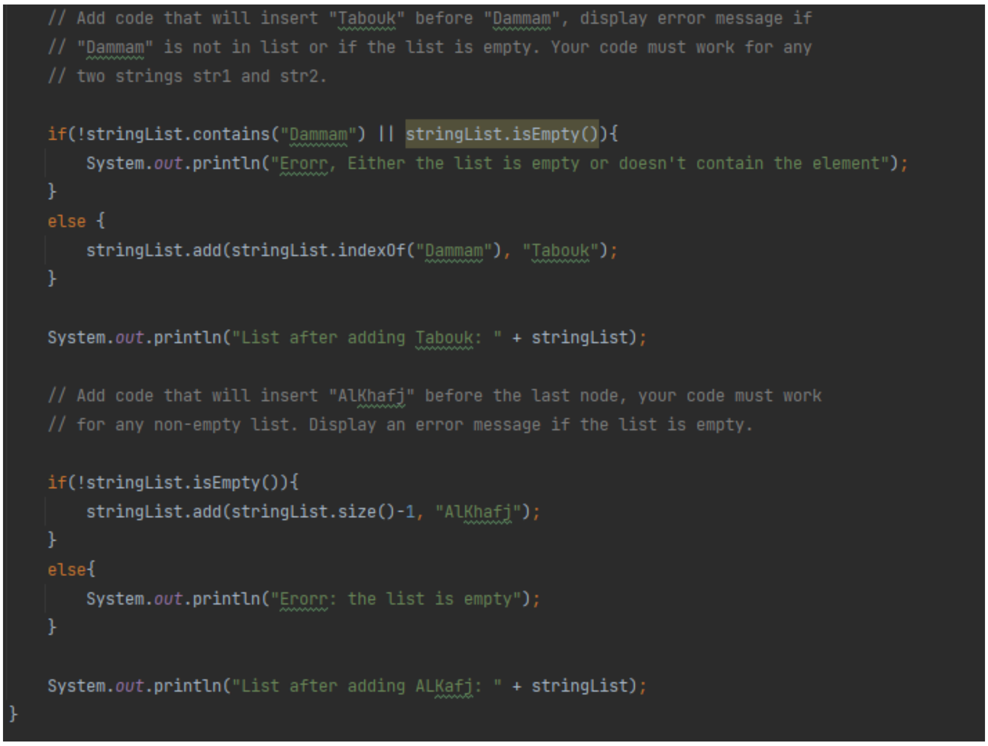
ICS-202 Lab-02 Report

Mohammed Busaleh - 202158210

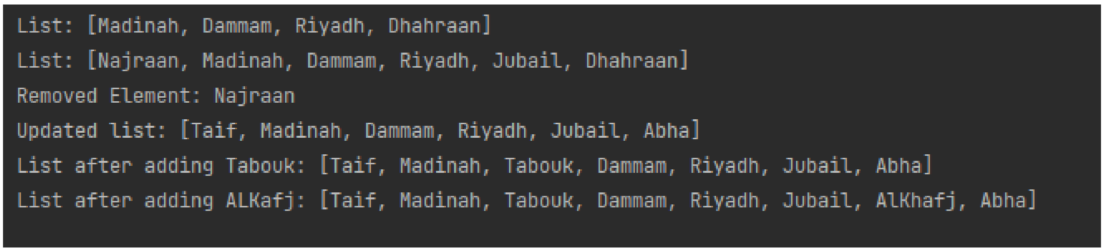
Task 1:

Code:



```
    // Add code that will insert "Tabouk" before "Dammam", display error message if
    // "Dammam" is not in list or if the list is empty. Your code must work for any
    // two strings str1 and str2.

    if(!stringList.contains("Dammam") || stringList.isEmpty()){
        System.out.println("Erorr, Either the list is empty or doesn't contain the element");
    }
    else {
        stringList.add(stringList.indexOf("Dammam"), "Tabouk");
    }

    System.out.println("List after adding Tabouk: " + stringList);

    // Add code that will insert "AlKhafj" before the last node, your code must work
    // for any non-empty list. Display an error message if the list is empty.

    if(!stringList.isEmpty()){
        stringList.add(stringList.size()-1, "AlKhafj");
    }
    else{
        System.out.println("Erorr: the list is empty");
    }

    System.out.println("List after adding ALKafj: " + stringList);
}
```

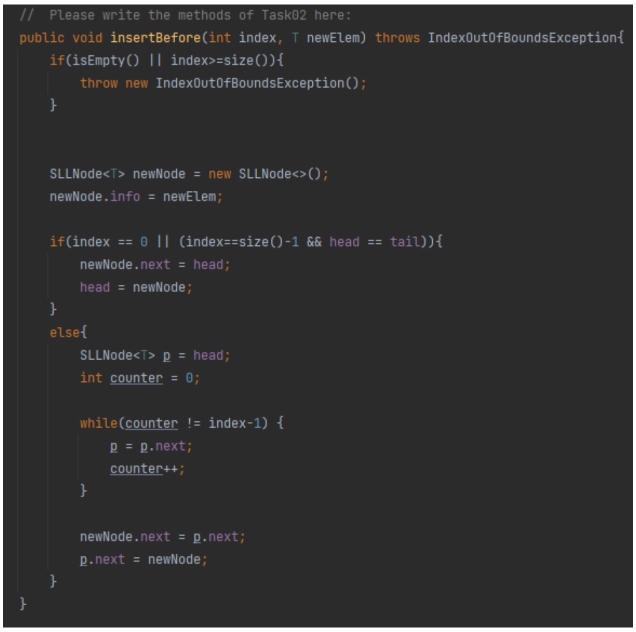
Output:



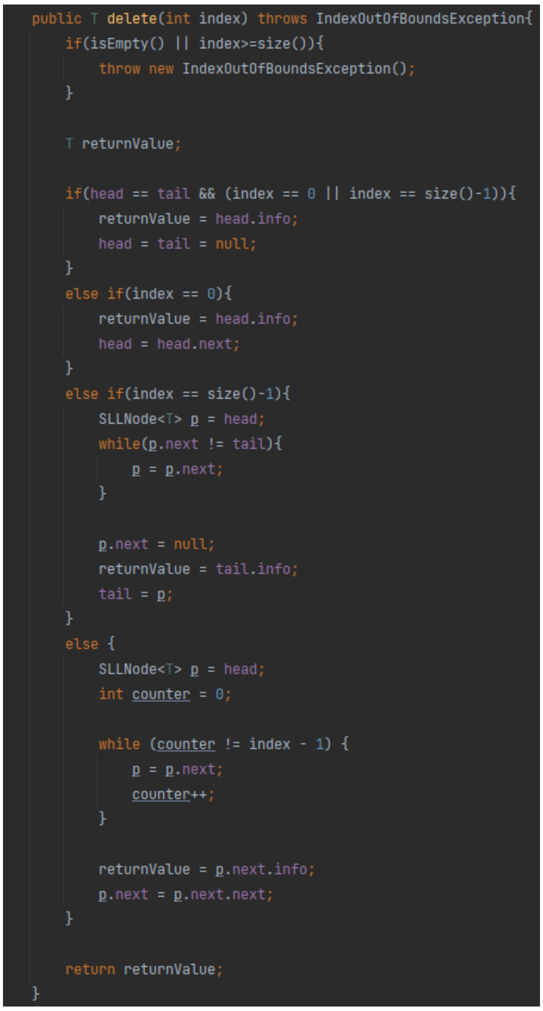
```
List: [Madinah, Dammam, Riyadh, Dhahraan]
List: [Najraan, Madinah, Dammam, Riyadh, Jubail, Dhahraan]
Removed Element: Najraan
Updated list: [Taif, Madinah, Dammam, Riyadh, Jubail, Abha]
List after adding Tabouk: [Taif, Madinah, Tabouk, Dammam, Riyadh, Jubail, Abha]
List after adding ALKafj: [Taif, Madinah, Tabouk, Dammam, Riyadh, Jubail, AlKhafj, Abha]
```

Task 2:

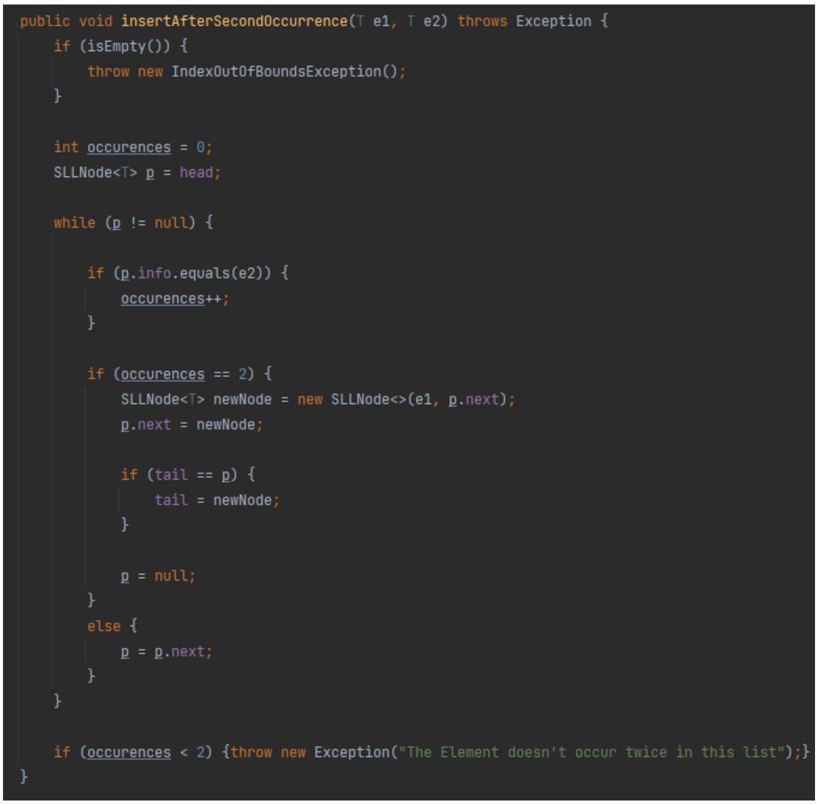
Code:



```java
//  Please write the methods of Task02 here:
public void insertBefore(int index, T newElem) throws IndexOutOfBoundsException{
    if(isEmpty() || index>=size()){
        throw new IndexOutOfBoundsException();
    }


    SLLNode<T> newNode = new SLLNode<>();
    newNode.info = newElem;

    if(index == 0 || (index==size()-1 && head == tail)){
        newNode.next = head;
        head = newNode;
    }
    else{
        SLLNode<T> p = head;
        int counter = 0;

        while(counter != index-1) {
            p = p.next;
            counter++;
        }

        newNode.next = p.next;
        p.next = newNode;
    }
}
```

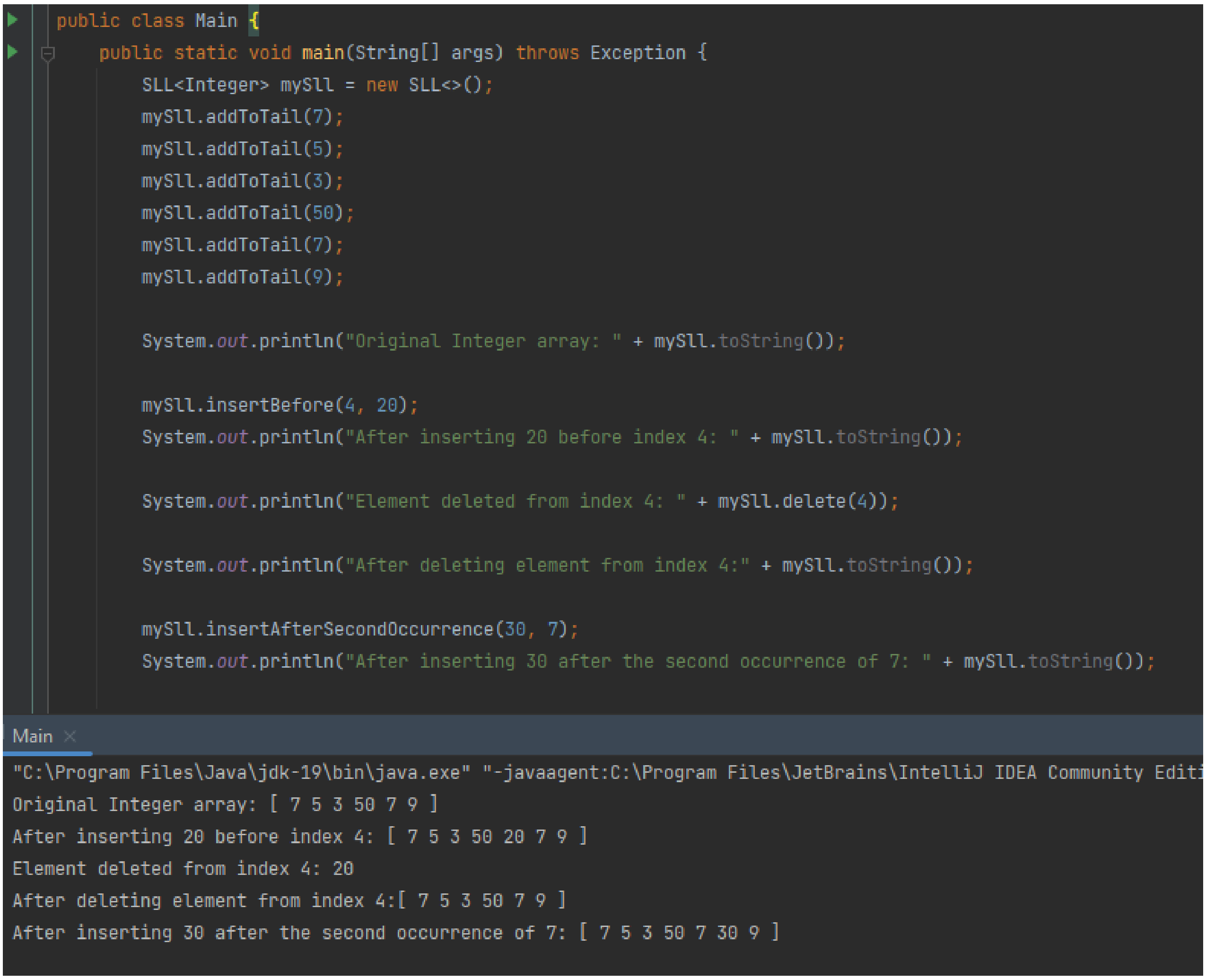


```java
public T delete(int index) throws IndexOutOfBoundsException{
    if(isEmpty() || index>=size()){
        throw new IndexOutOfBoundsException();
    }

    T returnValue;

    if(head == tail && (index == 0 || index == size()-1)){
        returnValue = head.info;
        head = tail = null;
    }
    else if(index == 0){
        returnValue = head.info;
        head = head.next;
    }
    else if(index == size()-1){
        SLLNode<T> p = head;
        while(p.next != tail){
            p = p.next;
        }

        p.next = null;
        returnValue = tail.info;
        tail = p;
    }
    else {
        SLLNode<T> p = head;
        int counter = 0;

        while (counter != index - 1) {
            p = p.next;
            counter++;
        }

        returnValue = p.next.info;
        p.next = p.next.next;
    }

    return returnValue;
}
```



```java
public void insertAfterSecondOccurrence(T e1, T e2) throws Exception {
    if (isEmpty()) {
        throw new IndexOutOfBoundsException();
    }

    int occurences = 0;
    SLLNode<T> p = head;

    while (p != null) {

        if (p.info.equals(e2)) {
            occurences++;
        }

        if (occurences == 2) {
            SLLNode<T> newNode = new SLLNode<>(e1, p.next);
            p.next = newNode;

            if (tail == p) {
                tail = newNode;
            }

            p = null;
        }
        else {
            p = p.next;
        }
    }

    if (occurences < 2) {throw new Exception("The Element doesn't occur twice in this list");}
}
```

Task 03:



```java
public class Main {
    public static void main(String[] args) throws Exception {
        SLL<Integer> mySll = new SLL<>();
        mySll.addToTail(7);
        mySll.addToTail(5);
        mySll.addToTail(3);
        mySll.addToTail(50);
        mySll.addToTail(7);
        mySll.addToTail(9);

        System.out.println("Original Integer array: " + mySll.toString());

        mySll.insertBefore(4, 20);
        System.out.println("After inserting 20 before index 4: " + mySll.toString());

        System.out.println("Element deleted from index 4: " + mySll.delete(4));

        System.out.println("After deleting element from index 4:" + mySll.toString());

        mySll.insertAfterSecondOccurrence(30, 7);
        System.out.println("After inserting 30 after the second occurrence of 7: " + mySll.toString());
```

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editi
Original Integer array: [ 7 5 3 50 7 9 ]
After inserting 20 before index 4: [ 7 5 3 50 20 7 9 ]
Element deleted from index 4: 20
After deleting element from index 4:[ 7 5 3 50 7 9 ]
After inserting 30 after the second occurrence of 7: [ 7 5 3 50 7 30 9 ]
```