DIT181- Data Structure and Algorithms Assignment 2

Question 1:

We create two stacks for inbox and one for outbox, then using enqueue that will push item inside the box and dequeue to push items outside the box and keep it always not empty by pushing items for inbox to outbox.

Dequeue has O(1) when it is not pushing items and O(n) when it is pushing items

Enqueue has O(1)

```
public class First_question<E> {
 private Stack<E> in = new Stack<E>();
 private Stack<E> out = new Stack<E>();

 public void enqueue(E item) {
   inbox.push(item);
 }
 public E dequeue() {
   if (out.isEmpty()) {
     while (!in.isEmpty()) {
       out.push(in.pop());
     }
   }
   return out.pop();
 }
```

Question 2:

Please see RPN.java file

Question 3:

Please see SinglyLinkedList.java file

The complexity of this method is: O(n)

Question 4:

Please see SinglyLinkedList.java file

The complexity of this method is: O(n)

Question 5:

Please see SinglyLinkedList.java file

The complexity of this method is: O(n)

Question 6:

Please see SinglyLinkedList.java file

The complexity of this method is: O(n)

Question 7:

Please see Tree.java file.

Question 8:

Please see Tree.java file.

Question 9:

Please see Tree.java file.