Lab6: Self Check

1) The big-O for a single linked list get operation get operation is O(N) as it must traverse through up until it finds the target node.

2) The big-O for a single linked list set operation is similar to the get big-O as it must traverse through the list to get to the point that wishes to be set therefore O(N).

3) The big-O for a single linked list add method depends on the situation. If the user gives the index the list must be traversed through making it O(N). If the user simply wants to add to the beginning or end the tail and head are already taken track of so the references can be changed without any traversing making the big-O O(1) in some cases for adding

4)
```
[5] -> [10] -> [7] -> [30] -> null
head                  tail

Node<Integer> nodeRef = head;
…
int next = nodeRef.data
…
nodeRef = nodeRef.next
```

5) a. The statement sets the head data to “shakira” and uses the previous head.next as the new head.next essentially ridding/replacing the original head.
b. Creates string node using the head’s reference, sets the next reference to the head.next’s next. Essentially adds node after the head
c. Traverses through the Linked List and adds new node at the end with data “tamika” - adds to end
d. Travereses list up to node with “Harry data”, if it finds “Harry” node it adds “sally” node before “Harry” by creating a new “sally” node and then setting “sally” next to “Harry” amd “Harry” next to “sally” next.next (2 after).