QUESTION 1

//a. In this code fragment, we want to remove the node before “Sam” (or before the tail node). This is achieved by first creating a temporal node reference called nodeRef and making this reference point to the node before the tail (or before Sam). Thus, we let nodeRef point to “Harry”. Afterwards, we let the previous of nodeRef (i.e. Dick) point to the tail node (i.e. Sam). Finally, we will let the previous pointer of tail (or previous pointer of Sam) point to the previous of nodeRef or “Dick”. This will remove “Harry” from the list and updated list is: Tom, Dick, Sam in the double-linked list in that order.

//b. In this code fragment, the goal is to create a new node with the data item “Tamika” and insert it in the beginning of the double-linked list so that the new node becomes the head. This is achieved by creating a String temporary node reference called nodeRef and making it point to the head. Next, create a new node with data item “Tamika” in it. Now let the head point to this new node. Next, let the next pointer of head or “Tamika” point to nodeRef or Tom. Finally, let the previous pointer of nodeRef or Tom point to the head which is “Tamika”. This will make the updated list: Tamika, Tom, Dick, Sam in that order.

//c. The goal of this code fragment is to create a new node with data item “Shakira” and insert it before Tom. We will first create a new node with the data item “Shakira” and let a temporal node pointer nodeRef point to this new node. Now, let the previous pointer of nodeRef (i.e. Shakira) points to head which is “Tamika”. Next, let the next pointer of nodeRef point to the next node of head which is “Tom”. Next, let the previous pointer of the next node after head (i.e. Tom) point to nodeRef (i.e. Shakira). Finally, let the head (i.e. Tamika) point to nodeRef or Shakira. These will result in the update list: Tamika, Shakira, Tom, Dick, Sam in that order.