


**WIA1002/WIB1002 Data Structures****Tutorial: Linked List & Doubly Linked List****Question 1**

```
1 public E xyz(int index,E e)
2 {
3     Node<E> current=head;
4     Node<E> temp;
5     if(index<0) return null;
6     else if(index>=size-1) {
7         this.addLast(e);
8         return null;
9     }
10    else if(index==0) {
11        temp=head;
12        head.element=e;
13        return temp.element;
14    }else{
15        for (int i = 1; i < index; i++) {
16            current=current.next;
17        }
18        temp=current.next;
19        current.next.element=e;
20        return temp.element;
21    }
22 }
```

Given method xyz with 2 arguments:

- a) Based on the above source code, explain what the lines of code do from line 10 – 21.

Based on the above source code, from line 10 – 13 is executed when then index is equal to 0, it replaces the head node's element with the new element e to the first index of the linked list. From line 14 – 21 is executed when storing the new element, e and locate where to replaces it by traversing the node. The new node will be replaced between the nodes current and temp. No new node is created or assigned but only updating the existing node's element.

- b) What is the main purpose of the method xyz()?

The main purpose of the method xyz() is to set an new element into the list at the specified index and returning the old value that have been replaced with.

- c) Obviously, there are some bugs in the source code. Debug the code and make it concise and simpler.

```
public E xyz(int index, E e){
    if (index < 0) {
        throw new IndexOutOfBoundsException("Index can not be negative!");
    }
    Node<E> current = this.head;
    E element;

    // Handle index = 0 case
    if (index == 0) {
        // let element = oldElement of head element
        element = head.element;
        head.element = e;
        return element;
    }

    // Auto-expanding list by using null insertion
    while (size <= index) {
        addLast(null);
    }

    // tranversing the nodes
    for (int i = 0; i < index; i++) {
        current = current.next;
    }
    element = current.element;
    current.element = e;
    return element;
}
```

Given method xyz with 2 arguments:

- a) Based on the above source code, explain what the lines of code do from line 10 - 21.

```

10      //to set new value for head
11      else if(index==0) {
12          temp=head;
13          head.element=e;
14          return temp.element;
15      }else{
16          //iterate forward beginning at index 1 and stop at the
17          requested index (ignore index 0 because already cater by code
18          line 10-13
19
20          for (int i = 1; i < index; i++) {
21              current=current.next;
22          }
23          //copy current.next node to the temp
24          temp=current.next;
25          //Assign the value of e to the element of current.next node
26          current.next.element=e;
27          //return the old value that has been replaced with (using temp
28          node)
29          return temp.element;
30      }

```

- b) What are the main purposes of the method xyz()?

- To set new value based on the given index and
- To return the old value that has been replaced with

- c) Modify above source code to make it concise and simpler

```

public E set(int index,E e){
    Node<E> current=head;
    E element;
    if(index<0)return null;
    else if(index>=size-1){
        this.addLast(e);
        return null;
    }
    else{
        for (int i = 0; i < index; i++) {
            current=current.next;
        }
        element = current.element;
        current.element=e;
        return element;
    }
}

```

**Question 2**

```
1  else{
2      Node<E> temp = head;
3      for(int i=0; i<index; i++){
4          temp = temp.next;
5      }
6      element = temp.element;
7      temp.next.prev = temp.prev;
8      temp.prev.next = temp.next;
9      temp.next = null;
10     temp.prev = null;
11     size --;
12 }
```

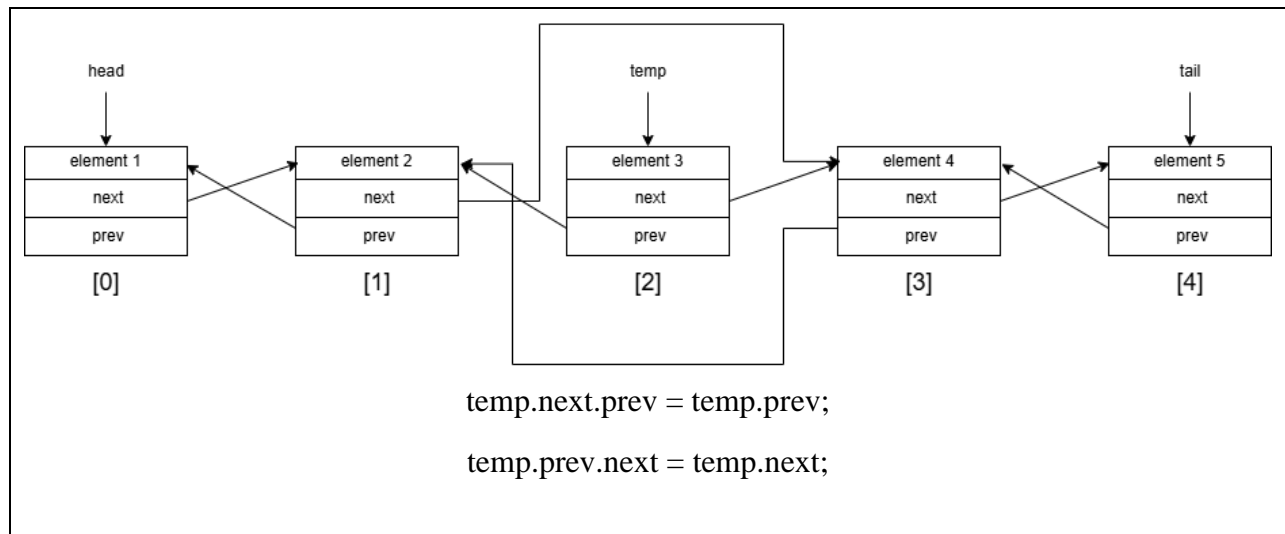
Based on the source code above, assume the index given is 3

- a) Explain what the lines of code do from line 2-11.

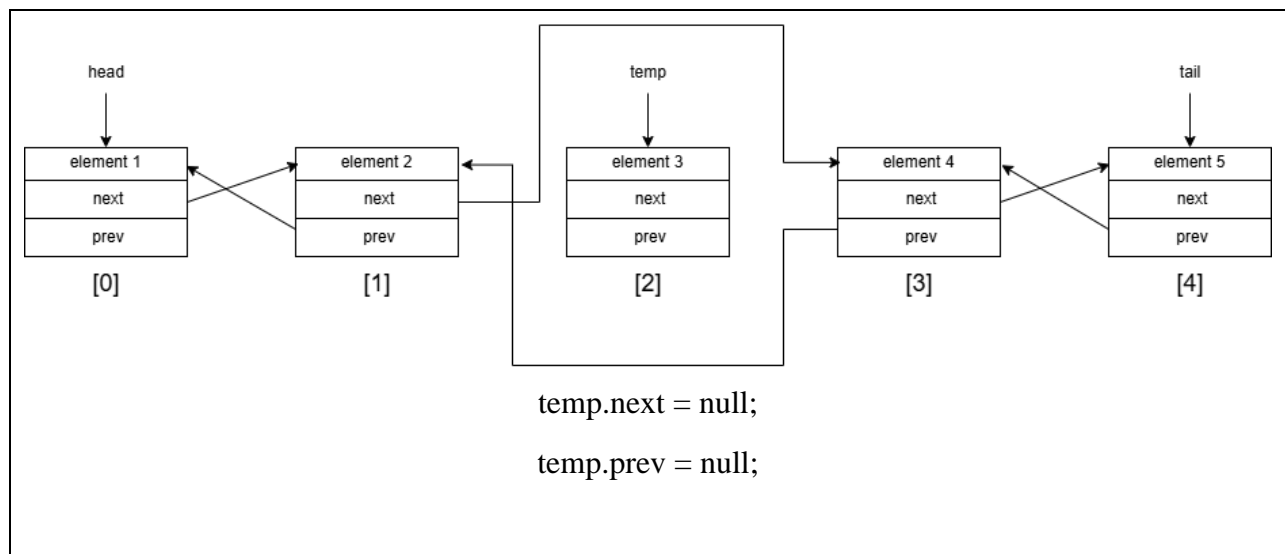
From line 2 to 11, the lines of code execute the removal of a node at a specified index in a doubly linked list. At line 2, a temporary node **temp** is initialized by pointing towards the **head** of the list. From line 3 to line 5, a loop is used to traverse the list until the **temp** points to the node at the targeted **index**. Line 6 assigns the **element** variable to **temp's stored element** once found. Line 7 updates the **next node's previous pointer** to point back to **temp's previous node** while Line 8 updates the **previous node's next** by pointing forward to **temp's next node**. Line 9 and 10 nullify **the temp variable's next and previous pointers** to fully disconnect it from the list and enabling garbage collection. Line 11 decrements the list **size** by 1 indicating that the success removal of a node in the list.

b) Draw the nodes for lines 7 – 10

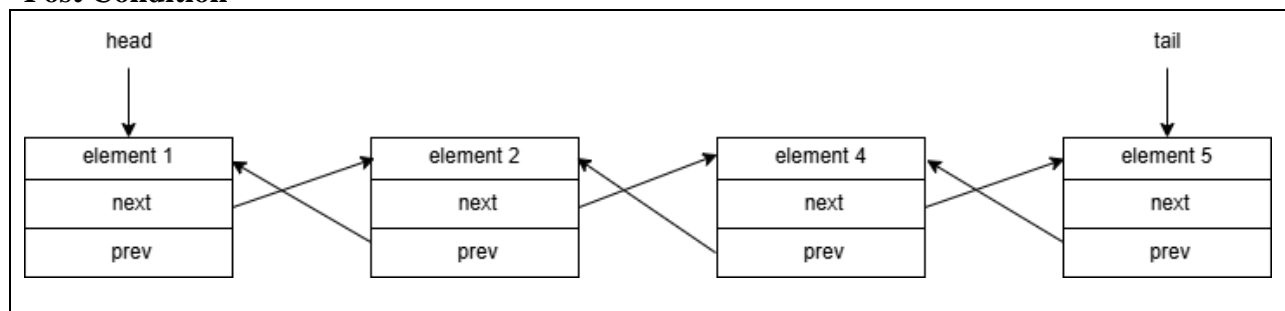
### Node condition at Line 7 and 8



### Node condition at Line 9 and 10



### Post Condition



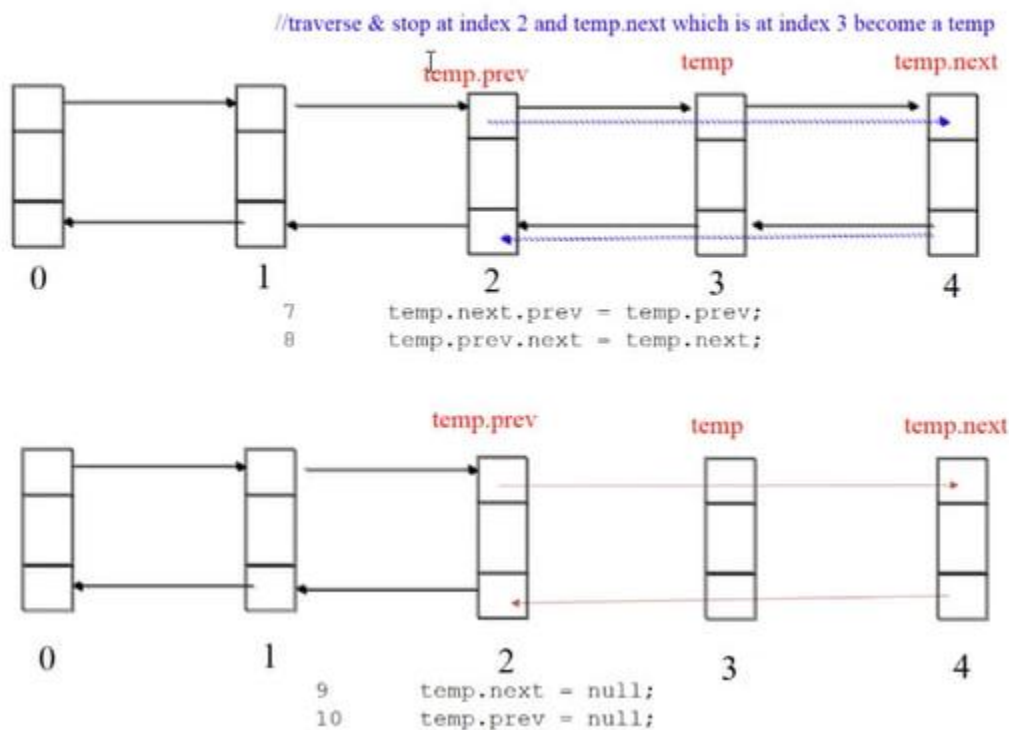
a) Explain what the lines of code do from line 2-12.

```

else{
    // create node temp and set point to the head
    Node<E> temp = head;
    //traverse and stop at the node of requested index
    for(int i=0; i<index; i++){
        temp = temp.next;
    }
    //copy element hold by temp node to the element
    element = temp.element;
    //repoint of pointer 'prev' of (temp.next) to the node
temp.prev
    temp.next.prev = temp.prev;
    //repoint of pointer 'next' of (temp.prev) to the node
temp.next
    temp.prev.next = temp.next;
    //set pointers of node temp (next and prev) to be null
    temp.next = null;
    temp.prev = null;
    //reduce number of node by one
    size --;
}

```

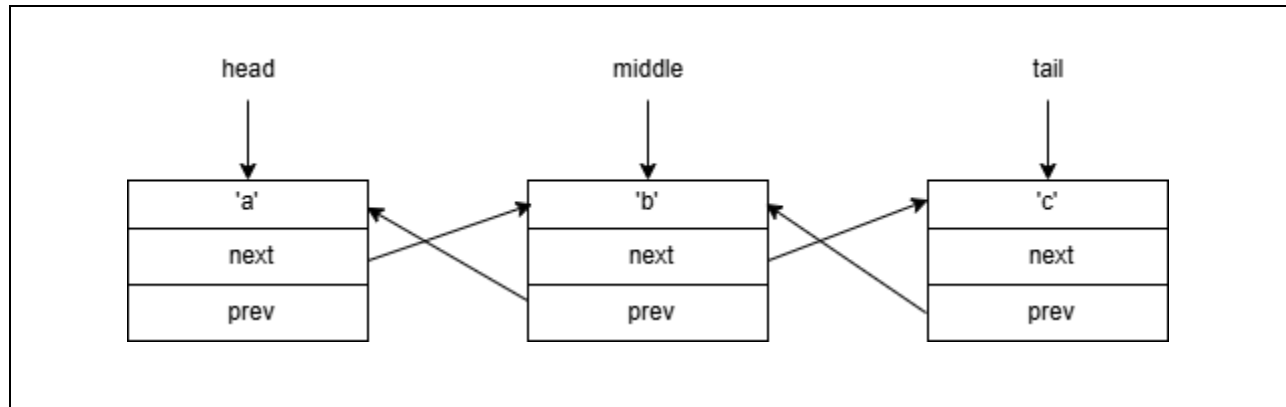
b) Draw the nodes for lines 7 - 10



**Question 3**

A doubly linked list keeps a set of characters. The head, the middle and the tail nodes respectively contains alphabet 'a', 'b' and 'c'. These nodes are in successive order. Create a new node that contains alphabet 'z'. Add this new node at the last location of this linked list. Draw all of these nodes including their correct references.

Before adding 'z'



After adding 'z'

