**Solve below Hacker Rank problem: -**

https://www.hackerrank.com/challenges/print-the-elements-of-a-linked-list/problem

**Finding Length of a LinkedList: -**

```
public class LinkedListLength {
	public static int getLinkedListLength(Node head) {
		if (head == null) {
			return 0;
		}
		Node p = head;
		int count = 0;
		while (p != null) {
			count++;
			p = p.next;
		}
		return count;
	}
}
```

**Insert First in LinkedList: -**

```
	public static Node insertFirst(Node head, int x) {
		Node p = new Node(x);
		if (head == null) {
			return p;
		}
		p.next = head;
		return p;
	}
```

**Insert Last in LinkedList: -**

```
	public static Node insertLast(Node head, int x) {
		Node p = new Node(x);
		if (head == null) {
			return p;
		}
		Node q = head;
		while (q.next != null) {
			q = q.next;
```

# Mercy Technologies

```
		}
		q.next = p;
		return head;
	}
```

**Insert Last in LinkedList: -**

```
	public static Node insertAtIndex(Node head, int x, int index) {
		int lenght = LinkedListLength.getLinkedListLength(head);
		Node updatedLinkedList = null;
		if (index > lenght) {
			return null;// You can throw an exception here!
		}
		if (index == 0) {
			updatedLinkedList = insertFirst(head, x);
			return updatedLinkedList;
		}
		if (index == lenght) {
			updatedLinkedList = insertLast(head, x);
			return updatedLinkedList;
		}
		Node p = head;
		Node q = new Node(x);
		int count = 1;
		while (p.next != null) {
			Node r = p.next;
			if (index == count) {
				p.next = q;
				q.next = r;
				break;
			}
			p=p.next;
			count++;
		}
		return head;
	}
```

**Hacker Rank Solution: -**

```
static SinglyLinkedListNode insertNodeAtPosition(SinglyLinkedListNode head, int da
ta, int position) {
    return insertAtIndex(head, data, position);
}
public static SinglyLinkedListNode insertFirst(SinglyLinkedListNode head, int x) {
    SinglyLinkedListNode p = new SinglyLinkedListNode(x);
    if (head == null) {
        return p;
    }
    p.next = head;
    return p;
}

public static SinglyLinkedListNode insertLast(SinglyLinkedListNode head, int x) {
    SinglyLinkedListNode p = new SinglyLinkedListNode(x);
```

# Mercy Technologies

```
    if (head == null) {
        return p;
    }
    SinglyLinkedListNode q = head;
    while (q.next != null) {
        q = q.next;
    }
    q.next = p;
    return head;
}

public static SinglyLinkedListNode insertAtIndex(SinglyLinkedListNode head, int x,
 int index) {
    int lenght = getLinkedListLength(head);
    SinglyLinkedListNode updatedLinkedList = null;
    if (index > lenght) {
        return null;// You can throw an exception here!
    }
    if (index == 0) {
        updatedLinkedList = insertFirst(head, x);
        return updatedLinkedList;
    }
    if (index == lenght) {
        updatedLinkedList = insertLast(head, x);
        return updatedLinkedList;
    }
    SinglyLinkedListNode p = head;
    SinglyLinkedListNode q = new SinglyLinkedListNode(x);
    int count = 1;
    while (p.next != null) {
        SinglyLinkedListNode r = p.next;
        if (index == count) {
            p.next = q;
            q.next = r;
            break;
        }
        p=p.next;
        count++;
    }
    return head;
}
//Not needed as per our Hacker Rank Problem.
public static int getLinkedListLength(SinglyLinkedListNode head) {
    if (head == null) {
        return 0;
    }
```

# Mercy Technologies

```
    SinglyLinkedListNode p = head;
    int count = 0;
    while (p != null) {
        count++;
        p = p.next;
    }
    return count;
}
```

**Solve below Hacker Rank problem: -**

https://www.hackerrank.com/challenges/insert-a-node-at-a-specific-position-in-a-linked-list/problem

https://www.hackerrank.com/challenges/insert-a-node-at-the-tail-of-a-linked-list/problem