NAME: Vidhi Gupta
Section: D
Batch: B2
Roll No: 22
Topic: Lab1:- Linked List

```
public class SinglyLinkedList {
   private Node head;

   public SinglyLinkedList() {
      head = null;
   }

   public void insert(int data) {
      Node newNode = new Node(data);

      if (head == null) {
         head = newNode;
         return;
      }

      Node current = head;
      while (current.next != null) {
         current = current.next;
      }
      current.next = newNode;
   }


   public void insertAtBeginning(int data) {
      Node newNode = new Node(data);
      newNode.next = head;
      head = newNode;
   }

   public void insertAtPosition(int data, int position) {
      if (position < 1) {
         System.out.println("Position should be greater than or equal to 1.");
         return;
      }

      Node newNode = new Node(data);

      if (position == 1) {
         insertAtBeginning(data);
         return;
```

```
    }

    Node current = head;
    int count = 1;
    while (current != null && count < position - 1) {
        current = current.next;
        count++;
    }

    if (current == null) {
        System.out.println("Position out of range.");
        return;
    }

    newNode.next = current.next;
    current.next = newNode;
}


public void deleteByKey(int key) {
    if (head == null) {
        System.out.println("The list is empty.");
        return;
    }

    if (head.data == key) {
        head = head.next;
        return;
    }

    Node current = head;
    while (current.next != null && current.next.data != key) {
        current = current.next;
    }

    if (current.next == null) {
        System.out.println("Key " + key + " not found.");
        return;
    }

    current.next = current.next.next;
}
public void printList() {
    if (head == null) {
        System.out.println("The list is empty.");
        return;
    }
```

```
    Node current = head;
    while (current != null) {
        System.out.print(current.data + " -> ");
        current = current.next;
    }
    System.out.println("null");
}

public static void main(String[] args) {
    SinglyLinkedList list = new SinglyLinkedList();

    list.insert(10);
    list.insert(20);
    list.insert(30);
    list.insert(40);
    list.insert(50);

    System.out.println("Original List:");
    list.printList();
    list.insertAtBeginning(5);

    System.out.println("List after inserting 5 at the beginning:");
    list.printList();
    list.insertAtPosition(25, 3);

    System.out.println("List after inserting 25 at position 3:");
    list.printList();
    list.insertAtPosition(60, 100);

    System.out.println("Trying to insert 60 at position 100:");
    list.printList();


    list.deleteByKey(40);
    System.out.println("List after deleting 40:");
    list.printList();

    list.deleteByKey(50);
    System.out.println("List after deleting 50:");
    list.printList();

    list.deleteByKey(20);
    System.out.println("List after deleting 20:");
    list.printList();

    list.deleteByKey(10);
```

```
        System.out.println("List after deleting 10:");
        list.printList();
    }

    class Node {
        int data;
        Node next;

        public Node(int data) {
            this.data = data;
            this.next = null;
        }
    }
}
```

OUTPUT:

```
Original List:
10 -> 20 -> 30 -> 40 -> 50 -> null
List after inserting 5 at the beginning:
5 -> 10 -> 20 -> 30 -> 40 -> 50 -> null
List after inserting 25 at position 3:
5 -> 10 -> 25 -> 20 -> 30 -> 40 -> 50 -> null
Position out of range.
Trying to insert 60 at position 100:
5 -> 10 -> 25 -> 20 -> 30 -> 40 -> 50 -> null
List after deleting 40:
5 -> 10 -> 25 -> 20 -> 30 -> 50 -> null
List after deleting 50:
5 -> 10 -> 25 -> 20 -> 30 -> null
List after deleting 20:
5 -> 10 -> 25 -> 30 -> null
List after deleting 10:
5 -> 25 -> 30 -> null


...Program finished with exit code 0
Press ENTER to exit console.
```

GitHub Link:

https://github.com/VidhiGupta75/DSA_Linkedlist/new/main