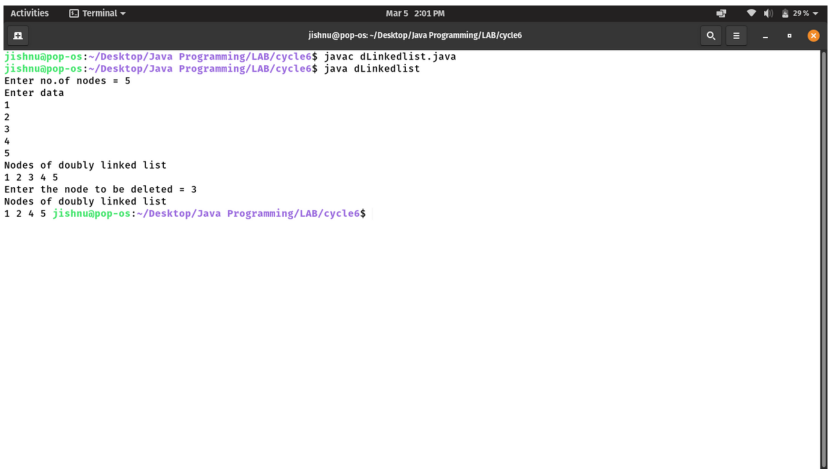
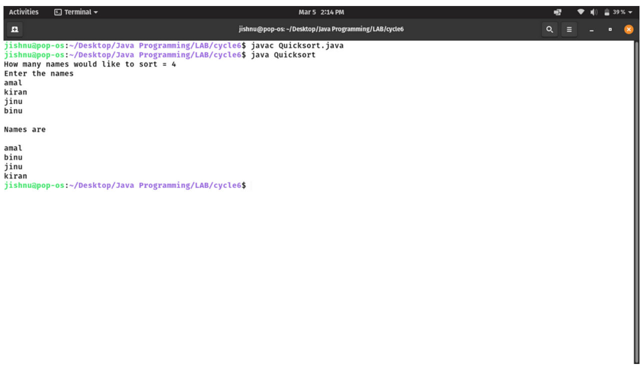
```
import java.util.*;
class DoublyLinkedList {
        class Node{
                int data;
                Node previous;
                Node next;
                public Node(int d) {
                        data = d;
                }
        }
        Node head, tail = null;
        public void addNode(int data) {
                Node newNode = new Node(data);
                        if(head == null) {
                                head = tail = newNode;
                                head.previous = null;
                                tail.next = null;
                        }else {
                                tail.next = newNode;
                                newNode.previous = tail;
                                tail = newNode;
                                tail.next = null;
                        }
        }
        public void deleteNode(int d){
                Node current=head;
                Node p=null;
                if(head == null) {
                        System.out.println("List is empty");
                        return;
                }
                while(current !=null && current.data !=d){
                        current=current.next;
                }
                if(current == null){
                        System.out.println("Given node is not present in the list");
                }else{
                        if(current.next!=null){
                                current.next.previous=current.previous;
                        }else{
                                tail=current.previous;
                        }
                        if(current.previous !=null){
                                current.previous.next=current.next;
                        }else{
                                head=current.next;
                        }
                }
        }
public void display() {
        Node current = head;
        if(head == null) {
                System.out.println("List is empty");
                return;
        }
        System.out.println("Nodes of doubly linked list");
        while(current != null) {
                System.out.print(current.data + " ");
                current = current.next;
} } }

class dLinkedlist{
        public static void main(String[] args) {
                DoublyLinkedList dList = new DoublyLinkedList();
                Scanner sc=new Scanner(System.in);
                System.out.print("Enter no.of nodes = ");
                int n=sc.nextInt();
                int temp,i=0;
                System.out.println("Enter data");
                while(i<n){
                        temp=sc.nextInt();
                        dList.addNode(temp);
                        i++;
                }
                dList.display();
                System.out.print("\nEnter the node to be deleted = ");
                temp=sc.nextInt();
                dList.deleteNode(temp);
                dList.display();
}}
```



```
jishnu@pop-os:~/Desktop/Java Programming/LAB/cycle6$ javac dLinkedlist.java
jishnu@pop-os:~/Desktop/Java Programming/LAB/cycle6$ java dLinkedlist
Enter no.of nodes = 5
Enter data
1
2
3
4
5
Nodes of doubly linked list
1 2 3 4 5
Enter the node to be deleted = 3
Nodes of doubly linked list
1 2 4 5 jishnu@pop-os:~/Desktop/Java Programming/LAB/cycle6$
```

```
import java.util.*;
public class Quicksort{
        String names[];
        int length;
        public static void main(String[] args) {
                Quicksort obj = new Quicksort();
                Scanner sc=new Scanner(System.in);
                System.out.print("How many names would like to sort = ");
                int temp=sc.nextInt();
                System.out.println("Enter the names");
                int k=0;
                String stringsList[]=new String[temp+1];
                while(k<=temp){
                        stringsList[k] = sc.nextLine();
                        k++;
                }
                obj.sort(stringsList);
                System.out.println("\nNames are");
                for (String i : stringsList) {
                        System.out.print(i);
                        System.out.println("");
                }
        }
        void sort(String array[]) {
                if (array == null || array.length == 0) {
                return;
        }
        this.names = array;
        this.length = array.length;
        quickSort(0, length - 1);
}
void quickSort(int lowerIndex, int higherIndex) {
        int i = lowerIndex;
        int j = higherIndex;
        String pivot = this.names[lowerIndex + (higherIndex - lowerIndex) / 2];
        while (i <= j) {
                while (this.names[i].compareToIgnoreCase(pivot) < 0) {
                        i++;
                }
                while (this.names[j].compareToIgnoreCase(pivot) > 0) {
                        j--;
                }
                if (i <= j) {
                        exchangeNames(i, j);
                                i++;
                                j--;
                }
        }
        if (lowerIndex < j) {
                quickSort(lowerIndex, j);
        }
        if (i < higherIndex) {
                quickSort(i, higherIndex);
        }
}
void exchangeNames(int i, int j) {
        String temp = this.names[i];
        this.names[i] = this.names[j];
        this.names[j] = temp;
}}
```



```
jishnu@pop-os:~/Desktop/Java Programming/LAB/cycle6$ javac Quicksort.java
jishnu@pop-os:~/Desktop/Java Programming/LAB/cycle6$ java Quicksort
How many names would like to sort = 4
Enter the names
amal
kiran
jinu
binu

Names are

amal
binu
jinu
kiran
jishnu@pop-os:~/Desktop/Java Programming/LAB/cycle6$
```