Name of Student :- Pavan Ratan Gole

UID:- Na

Batch:- C4

Exp. No. 4

Aim:- Given a Doubly Linked List of integers,
1- Remove all duplicates from the linked List
2-write a function to modify the linked list such that all even numbers appear before all the odd numbers in the modified linked list. Also, keep the order of even and odd numbers the same.

Program:-

```
import java.util.Scanner;


/**
* doublyLinkedlist
*/

class Node {
   Node prev;
   Node next;
   int data;

   Node(int data) {
       this.data = data;
   }
}

public class DoublyLinkedlist {
   Node head = null;
   Node tail = null;

   void insert(int data) {
       Node node = new Node(data);
```

```
        if(head == null) {
            head = tail = node;
        }
        else {
            tail.next = node;
            node.prev = tail;
            tail = node;
        }
    }

    void rmDuplicates() {
        Node temp = head;
        Node itr = head.next;
        while(temp != null) {
            itr = temp.next;
            while(itr != null) {
                if(temp.data == itr.data) {
                    if(itr == tail) {
                        itr.prev.next = null;
                        itr.prev = null;
                    }
                    else {
                        itr.prev.next = itr.next;
                        itr.next.prev = itr.prev;
                    }

                }
                itr = itr.next;
            }
            temp = temp.next;
        }
    }

    void sort(int totalodd) {
        int count = 1;
        Node dont = null;
        Node temp = head;
        Node crt = tail;
        while(head.data % 2 != 0) {
            if(totalodd == 0) {
```

```
                break;
            }
            Node cur = head;
            head = head.next;
            cur.next = null;
            cur.prev = tail;
            tail.next = cur;
            tail = cur;
            totalodd--;
        }
        crt = tail;
        temp = head;
        while(temp != null && temp.next != null) {
            if(head.data % 2 == 0) {
                if(temp.data % 2 != 0) {
                    if(totalodd == 0) {
                        break;
                    }
                    totalodd--;
                    if(temp == dont) {
                        break;
                    }
                    temp.prev.next = temp.next;
                    temp.next.prev = temp.prev;
                    Node cur = temp;

                    if(count == 1) {
                        dont = cur;
                        count++;
                    }

                    temp = temp.next;
                    cur.next = null;
                    crt.next = cur;
                    cur.prev = crt;
                    crt = cur;
                    if(temp.prev == tail) {
                        break;
                    }
                }
```

```
            else {
                if(temp == tail) {
                    break;
                }
                temp = temp.next;
            }
        }
        else {
            if(temp.data % 2 != 0) {
                if(totalodd == 0) {
                    break;
                }
                totalodd--;
                if(temp == dont) {
                    break;
                }
                temp.prev.next = temp.next;
                temp.next.prev = temp.prev;
                Node cur = temp;

                if(count == 1) {
                    dont = cur;
                    count++;
                }

                temp = temp.next;
                cur.next = null;
                crt.next = cur;
                cur.prev = crt;
                crt = cur;
                if(temp.prev == tail) {
                    break;
                }
            }
            else {
                if(temp == tail) {
                    break;
                }
                temp = temp.next;
            }
```

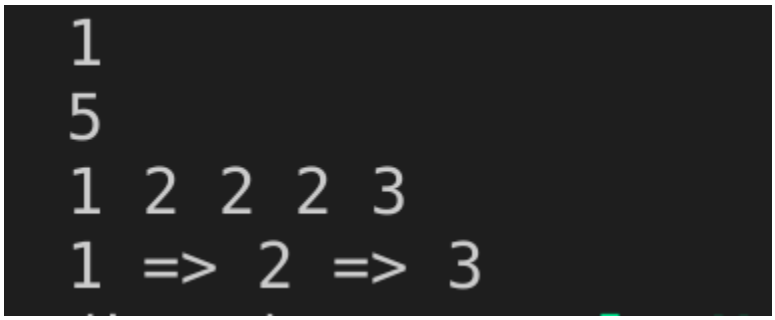
```
            }
        }
    }
    void display() {
        Node temp = head;
        while(temp != null) {
            if(temp.next != null) {
                System.out.print(temp.data + " => ");
            }
            else
                System.out.print(temp.data);
            temp = temp.next;
        }
        System.out.println();
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int countodd = 0;
        int t = sc.nextInt();
        {
            while(t-- > 0) {
                DoublyLinkedlist list = new DoublyLinkedlist();
                int n = sc.nextInt();
                for (int i = 0; i < n; i++) {
                    int input = sc.nextInt();
                    if(input % 2 != 0) {
                        countodd ++;
                    }
                    list.insert(input);
                }
                // list.sort(countodd);
                list.rmDuplicates();
                list.display();
               list.sort();
            }
        }
    }
}
```

Output:-

1) **Remove Duplicates**



2) **Odd after Even**