Code:-

public class linked\_list {

static class ListNode {

int val;

ListNode next;

ListNode(int v) {

val = v;

}

}

static class SinglyLinkedList {

public ListNode head;

public ListNode tail;

public int size;

public SinglyLinkedList() {

head = null;

tail = null;

size = 0;

}

public void addEnd(int x) {

ListNode n = new ListNode(x);

if (head == null) {

head = tail = n;

} else {

tail.next = n;

tail = n;

}

size++;

}

public void addAfter(int a, int b) {

ListNode temp = head;

while (temp != null && temp.val != a) {

temp = temp.next;

}

if (temp != null) {

ListNode n = new ListNode(b);

n.next = temp.next;

temp.next = n;

if (temp == tail) {

tail = n;

}

size++;

}

}

public void deleteFirst(int x) {

if (head == null) {

return;

}

if (head.val == x) {

head = head.next;

if (head == null) {

tail = null;

}

size--;

return;

}

ListNode prev = head;

ListNode cur = head.next;

while (cur != null) {

if (cur.val == x) {

prev.next = cur.next;

if (cur == tail) {

tail = prev;

}

size--;

return;

}

prev = cur;

cur = cur.next;

}

}

public void reverseK(int k) {

if (k <= 1 || head == null) {

return;

}

ListNode dummy = new ListNode(0);

dummy.next = head;

ListNode groupPrev = dummy;

while (true) {

ListNode kth = groupPrev;

for (int i = 0; i < k && kth != null; i++) {

kth = kth.next;

}

if (kth == null) {

break;

}

ListNode groupNext = kth.next;

ListNode prev = groupNext;

ListNode cur = groupPrev.next;

while (cur != groupNext) {

ListNode tmp = cur.next;

cur.next = prev;

prev = cur;

cur = tmp;

}

ListNode tmp = groupPrev.next;

groupPrev.next = kth;

groupPrev = tmp;

}

head = dummy.next;

tail = head;

if (tail != null) {

while (tail.next != null) {

tail = tail.next;

}

}

}

public void dedup() {

ListNode p = head;

while (p != null) {

ListNode prev = p;

ListNode r = p.next;

while (r != null) {

if (r.val == p.val) {

prev.next = r.next;

if (r == tail) {

tail = prev;

}

size--;

r = prev.next;

} else {

prev = r;

r = r.next;

}

}

p = p.next;

}

}

public String toStringList() {

StringBuilder sb = new StringBuilder();

ListNode cur = head;

while (cur != null) {

if (sb.length() > 0) {

sb.append(' ');

}

sb.append(cur.val);

cur = cur.next;

}

return sb.toString();

}

public int size() {

return size;

}

}

public static void main(String[] args) {

SinglyLinkedList pl = new SinglyLinkedList();

pl.addEnd(10);

pl.addEnd(20);

pl.addAfter(10, 15);

pl.addEnd(10);

pl.dedup();

pl.reverseK(2);

System.out.println(pl.toStringList());

}

}

Output:-

