In-Class Exercise

Practice Problem:

Write a function that removes the element of a given value in a linked list.

Assumptions:

- Node values are unique
- The value is never in the head node (the head/first node will never be deleted).
- The linked list will have at least two elements.

Given linked list -- head = [1,2,3] which looks like the following:

Example 1:

Input: value = 2
Output: [1,3]
```java
public class Node {
    int item;
    Node next;
    // Node Constructor
    Node(int d) {
        item = d;
        next = null;
    }
}

public class Tester{
    public static void main(String[] args){
        Node head = new Node(1);
        Node second = new Node(2);
        Node third = new Node(3);
        head.next = second;
        second.next = third;

        /* The current linked list is as follows:
        head | second | third
        | +--------+ +--------+ +--------+
        | 1       | 2       | 3       |
        | o--------o o--------o o--------o */

        System.out.println("List Before Deletion");
        printLinkedList(head);

        // User wants to delete value 2
        deleteValue(2, head);
        System.out.println("List After Deletion of 2");
        printLinkedList(head);

        // User wants to delete value 1
        deleteValue(3, head);
        System.out.println("List After Deletion of 3");
        printLinkedList(head);
    }

    // To pass the linked list to a function, you only need to pass the head
    public static void deleteValue(int value, Node head){
        Node prev, cur;
        for( cur = head, prev = null; cur!=null; prev = cur, cur = cur.next){
            if(cur.item == value){
                // Proceed to delete the node of interest
                prev.next = cur.next;
            }
            break; // Break the loop
        }
    }
}
```
// Node traversal and printing
public static void printLinkedList(Node head){
    for(Node cur = head; cur!=null; cur=cur.next){
        System.out.print(cur.item+" ");
    }
    System.out.println();
}
}