Delete the node

There is a singly-linked list `head` and we want to delete a node `node` in it.

You are given the node to be deleted `node`. You will **not be given access** to the first node of `head`.

All the values of the linked list are **unique**, and it is guaranteed that the given node `node` is not the last node in the linked list.

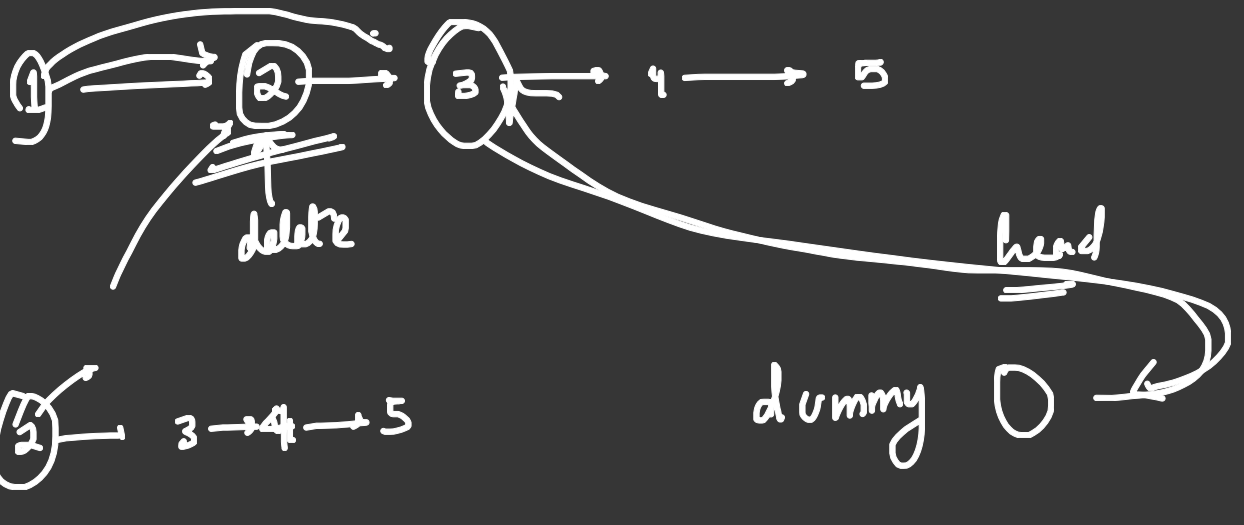
Delete the given node. Note that by deleting the node, we do not mean removing it from memory. We mean:

- The value of the given node should not exist in the linked list.
- The number of nodes in the linked list should decrease by one.
- All the values before `node` should be in the same order.
- All the values after `node` should be in the same order.

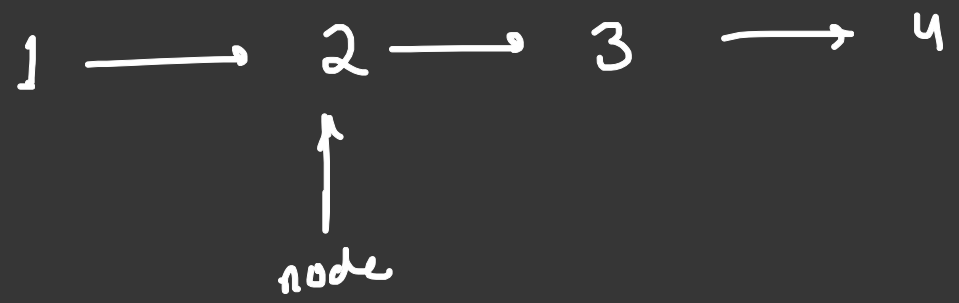
Custom testing:

- For the input, you should provide the entire linked list `head` and the node to be given `node`. `node` should not be the last node of the list and should be an actual node in the list.
- We will build the linked list and pass the node to your function.
- The output will be the entire list after calling your function.

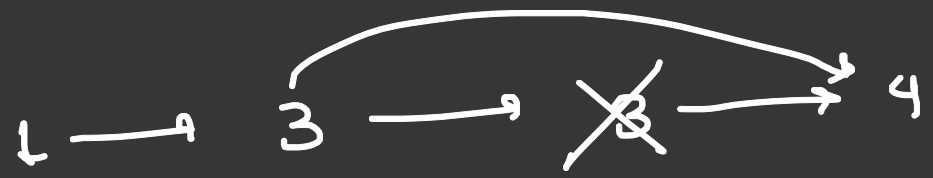
We have given directly the node which needs to be deleted.



- Idea to is to not to delete the node instead replacing the value with next node and connecting the node → next → next.



to node → data = node → next → data



node → next = node → next → next;

1 → 3 → 4

Pseudocode :

```
delete temp->data = temp->next->data;
    temp->next = temp->next->next;
    return;
```

T.C. = O(1)

S.C. = O(1)