Question 2

Assume a singly linked list with no access to head node or pointer.

Expectations

1. Design an algorithm to delete a node from the list where the node to be deleted is passed as an argument to the method

* Copy the data from the next node to the node to be deleted.
* Update the next pointer of the node to be deleted to skip the next node.

2. Write the code to implement the method deleteNode(Node n);

// Function to delete a node from the linked list

void deleteNode(struct Node\* n) {

// Check if the node is not the last node

if (n == NULL || n->next == NULL) {

printf("Cannot delete the last node using this method.\n");

return;

}

// Copy the data from the next node to the current node

struct Node\* temp = n->next;

n->data = temp->data;

// Update the next pointer to skip the next node

n->next = temp->next;

// Free the memory of the next node

free(temp);

}

3. List down any corner cases

The method should not be used to delete the last node, as it involves copying data from the next node which won't exist for the last node.

Ensure that the given node is not NULL.

if (n == NULL || n->next == NULL) {

printf("Cannot delete the last node using this method.\n");

return;

}

4. Complexity of the algorithm

The time complexity of this algorithm is O(1) because it involves constant-time operations, and the space complexity is also O(1) as no additional space is used.