# **Delete N Nodes After M Nodes**

**Task:** Given a linked list and two integers M and N. Traverse the linked list such that you retain M nodes then delete next N nodes, continue the same till end of the linked list.

**Examples:**

Input:

M = 2, N = 2

Linked List: 1->2->3->4->5->6->7->8

Output:

Linked List: 1->2->5->6

Input:

M = 3, N = 2

Linked List: 1->2->3->4->5->6->7->8->9->10

Output:

Linked List: 1->2->3->6->7->8

Input:

M = 1, N = 1

Linked List: 1->2->3->4->5->6->7->8->9->10

Output:

Linked List: 1->3->5->7->9

**Notes:**

* The main part of the problem is to maintain proper links between nodes, make sure that all corner cases are handled.
* Assume that M cannot be 0.

**Singles to Doubles**

**Task:** convert a single linked list to a double linked list

# **Linked Palindrome**

**Task:** Given a linked list of characters, recursively check if it is a palindrome or not

**Examples:**

Input: A —> B —> C —> B —> A —> null

Output: The linked list is a palindrome

Input: A —> B —> C —> C —> B —> null

Output: The linked list is not a palindrome

**Notes:** This can be done without constructing a string out of the characters

# **Reverse the List**

**Task:** Write a function that will reverse a linked list