**Lab Task: Reverse a Linked List Using Stack**

**Objective:**

Create a Python program that uses a **stack** to reverse a singly linked list.

**Problem Statement:**

Write a Python function reverse\_linked\_list(head: Node) -> Node that reverses a given singly linked list using a stack. You are given the head node of a singly linked list, and you need to return the head of the new reversed list.

**Constraints:**

* Implement the solution using a **stack** (do not use Python's built-in list reversing or reverse function).
* A singly linked list is a list of nodes where each node has a value and a reference to the next node.

**Function Signature:**

class Node:

def \_\_init\_\_(self, value=0, next=None):

self.value = value

self.next = next

def reverse\_linked\_list(head: Node) -> Node:

pass

**Steps to Implement:**

1. Traverse the given linked list and push all the node values onto a stack.
2. Pop the values from the stack and reconstruct the linked list in reverse order.
3. Return the head of the new reversed list.

**Example Inputs and Outputs:**

1. **Example 1:**
   * **Input:** 1 -> 2 -> 3 -> 4 -> 5
   * **Output:** 5 -> 4 -> 3 -> 2 -> 1
2. **Example 2:**
   * **Input:** 7 -> 8 -> 9
   * **Output:** 9 -> 8 -> 7
3. **Example 3:**
   * **Input:** 10
   * **Output:** 10
4. **Example 4:**
   * **Input:** None (Empty List)
   * **Output:** None