Lab Assignment: Intersection Point of Two Linked Lists

Objective

Implement a function in C to find the intersection point of two singly linked lists—if they intersect. Use a method that runs in $\mathcal{O}(n)$ time and uses $\mathcal{O}(1)$ extra space.

Problem Statement

Given two singly linked lists, determine the node at which they intersect. Return a pointer/reference to the intersecting node. If the lists do not intersect, return `NULL`.

Function Signature

```
struct Node* getIntersectionNode(struct Node* headA, struct Node* headB);
```

Optimized Approach

- Traverse both lists to compute their lengths m and n.
- Compute the length difference $diff = |m - n|$.
- Advance the pointer in the longer list by $diff$ nodes.
- Move both pointers (one from each list) one step at a time.
- The point at which both pointers meet is the intersection node. If they reach `NULL`, there is no intersection.

Example

List A: $1 \to 2 \to 3 \to 4 \to 5$
List B: $6 \to 4 \to 5$ (Here, node 4 onward is shared)
Intersection point = Node with value `4`