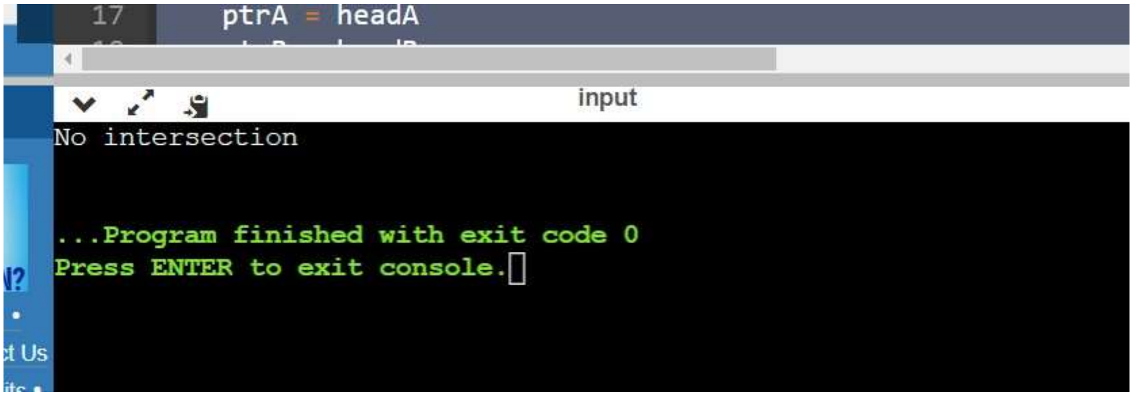
Day – 6 : Linked List -II

Problem 1: Given the heads of two singly linked-lists **headA** and **headB**, return **the node at which the two lists intersect.** If the two linked lists have no intersection at all, return **null.**

```
    class ListNode:
    def __init_(self, val=0, next=None):
        self.val = val
        self.next = next

def getIntersectionNode(headA, headB):
    def getLength(node):
        length = 0
        while node:
            length += 1
            node = node.next
        return length

    lenA = getLength(headA)
    lenB = getLength(headB)

    ptrA = headA
    ptrB = headB

    diff = abs(lenA - lenB)
    if lenA > lenB:
        for _ in range(diff):
            ptrA = ptrA.next
    else:
        for _ in range(diff):
            ptrB = ptrB.next
    while ptrA and ptrB:
        if ptrA == ptrB:
            return ptrA
        ptrA = ptrA.next
        ptrB = ptrB.next

    return None


def createLinkedList(lst):
    if not lst:
        return None
    head = ListNode(lst[0])
    current = head
    for i in range(1, len(lst)):
        current.next = ListNode(lst[i])
        current = current.next
    return head


list1 = createLinkedList([1, 3, 1, 2, 4])
list2 = createLinkedList([3, 2, 4])
intersection_node = getIntersectionNode(list1, list2)
if intersection_node:
    print(intersection_node.val)
else:
    print("No intersection")
```



```
    ptrA = headA
```

```
No intersection


...Program finished with exit code 0
Press ENTER to exit console.
```