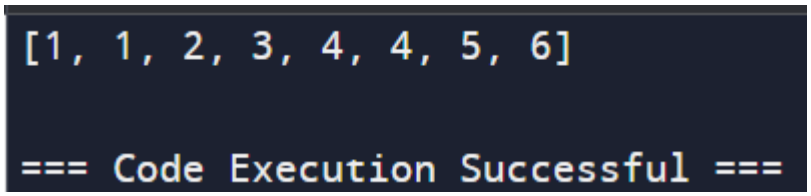
42. Merge k Sorted Lists You are given an array of k linked-lists lists, each linked-list is sorted in ascending order. Merge all the linked-lists into one sorted linked-list and return it. Example 1: Input: lists = [[1,4,5],[1,3,4],[2,6]] Output: [1,1,2,3,4,4,5,6] Explanation: The linked-lists are: [1->4->5, 1->3->4, 2->6] merging them into one sorted list: 1->1->2->3->4->4->5->6

PROGRAM:-

```
import heapq

# Definition for a singly-linked list node.
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

    # To help with heap comparison.
    def __lt__(self, other):
        return self.val < other.val

def mergeKLists(lists):
    # Initialize a priority queue (min-heap).
    heap = []

    # Push the head of each list into the heap.
    for l in lists:
        if l:
            heapq.heappush(heap, l)

    # Create a dummy node to serve as the starting point of the merged list.
    dummy = ListNode()
    current = dummy

    # Extract the smallest element from the heap, and then push the next element of that list into the heap.
    while heap:
        smallest_node = heapq.heappop(heap)
        current.next = smallest_node
        current = current.next
        if smallest_node.next:
            heapq.heappush(heap, smallest_node.next)

    # Return the merged list, which starts from dummy.next.
    return dummy.next

# Helper function to create linked lists from a list of lists.
def create_linked_lists(arrays):
    lists = []
    for arr in arrays:
        if arr:
            head = ListNode(arr[0])
            current = head
```

```
            for val in arr[1:]:
                current.next = ListNode(val)
                current = current.next
            lists.append(head)
        else:
            lists.append(None)
    return lists

# Helper function to print the linked list.
def print_linked_list(head):
    result = []
    while head:
        result.append(head.val)
        head = head.next
    print(result)

# Example usage:
lists = [[1,4,5], [1,3,4], [2,6]]
linked_lists = create_linked_lists(lists)
merged_head = mergeKLists(linked_lists)
print_linked_list(merged_head)
```

OUTPUT:-



TIME COMPLEXITY:-O(n log k)