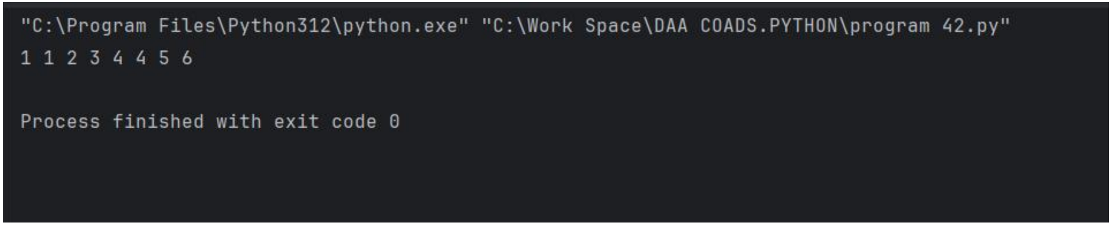
Program 42. Merge k Sorted Lists You are given an array of k linked-lists lists, each linked-list is sorted in ascending order. Merge all the linked-lists into one sorted linked-list and return it. Example 1: Input: lists = [[1,4,5],[1,3,4],[2,6]] Output: [1,1,2,3,4,4,5,6] Explanation: The linked-lists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

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

def mergeKLists(lists):
    heap = []
    dummy = ListNode()
    current = dummy

    # Push the first node of each list into the heap
    for i, node in enumerate(lists):
        if node:
            heapq.heappush(heap, (node.val, i, node))

    # Process the heap until empty
    while heap:
        val, i, node = heapq.heappop(heap)
        current.next = node
        current = current.next
        if node.next:
            heapq.heappush(heap, (node.next.val, i, node.next))

    return dummy.next

# Example usage
def printList(node):
    while node:
        print(node.val, end=" ")
        node = node.next
    print()

# Example input
lists = [
    ListNode(1, ListNode(4, ListNode(5))),
    ListNode(1, ListNode(3, ListNode(4))),
    ListNode(2, ListNode(6))
]
```

```
merged_head = mergeKLists(lists)
printList(merged_head)  # Output: 1 1 2 3 4 4 5 6
```

Output::



```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 42.py"
1 1 2 3 4 4 5 6

Process finished with exit code 0
```

Time complexity:
O(log k)