**Solve (Set#A)**

| **Python** | |
| --- | --- |
| def isAscending(head):  if not head or not head.next:  return True  if head.val >= head.next.val:  return False  return isAscending(head.next) | |
| **Java** | |
| **public boolean isAscendingOrder(ListNode head) {**  **if (head == null || head.next == null) {**  **return true;**  **}**  **if (head.val >= head.next.val) {**  **return false;**  **}**  **return isAscendingOrder(head.next);**  **}** | |

**Rubric (Set#A)**

| **SI** | **Category** | **Marks** |
| --- | --- | --- |
| 1 | Correct Base Condition | 3 |
| 2 | Correct Recursive Function call with parameter | 5 |
| 3 | Coherent Function return for True or False | 4 |
| 4 | Coherent Sequence of operations | 3 |
| **Total = 15** | | |

**Solve (Set # B)**

| **Python** | |
| --- | --- |
| def isDescendingOrder(head):  if not head or not head.next:  return True  if head.val <= head.next.val:  return False  return isDescendingOrder(head.next) | |
| **Java** | |
| **public boolean isDescendingOrder(ListNode head) {**  **if (head == null || head.next == null) {**  **return true;**  **}**  **if (head.val <= head.next.val) {**  **return false;**  **}**  **return isDescendingOrder(head.next);**  **}** | |

**Rubric (Set#B)**

| **SI** | **Category** | **Marks** |
| --- | --- | --- |
| 1 | Correct Base Condition | 3 |
| 2 | Correct Recursive Function call with parameter | 5 |
| 3 | Coherent Function return for True or False | 4 |
| 4 | Coherent Sequence of operations | 3 |
| **Total = 15** | | |