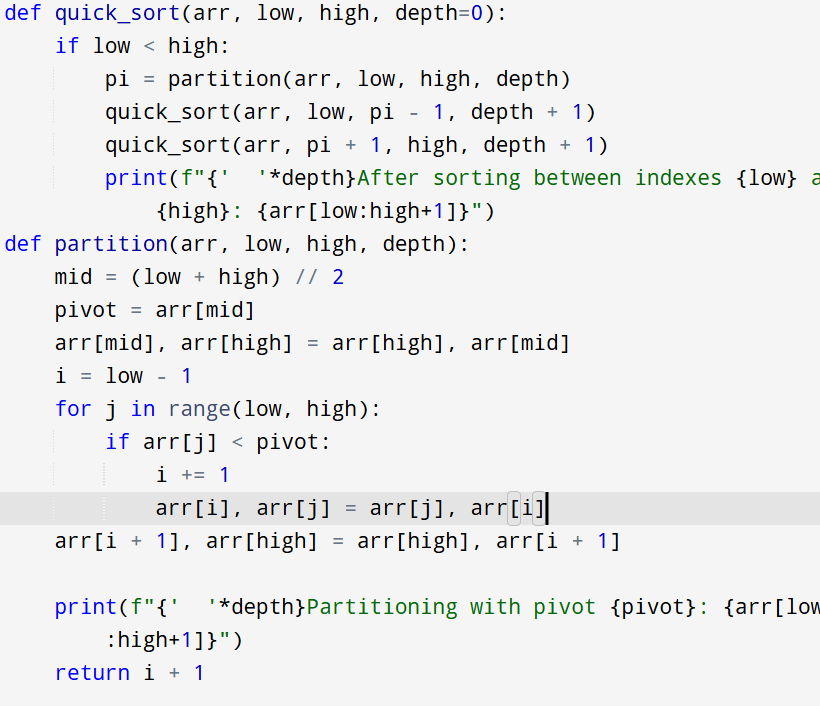
**3.6 QUICK SORTING**

**Aim**: To write a Python program to execute quick sort.

**Algorithm:**

1. Start
2. Input: Array a[] and its size N
3. If array has 0 or 1 elements, return (already sorted)
4. Select Pivot: Choose the middle element as pivot
5. Partition:
6. Place all elements less than pivot on the left
7. Place all elements greater than or equal to pivot on the right
8. Recursively apply Quick Sort on left and right sub-arrays
9. Output:
10. Show array after each partitioning/recursion
11. End

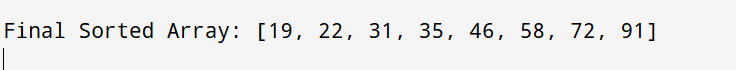
**Program:**

****

**Input:**

a = [19, 72, 35, 46, 58, 91, 22, 31]

**Output:**

****

**Result:** Thus the program is successfully executed and the output is verified.

**Performance analysis:**

|  |  |
| --- | --- |
| Best/Average Time | :O(n log n) |

|  |  |
| --- | --- |
| Worst Time (unbalanced pivot) | :O(n²) |

|  |  |
| --- | --- |
| Space Complexity | :O(log n) recursion stack |

|  |  |
| --- | --- |
|  |  |