**3.5** Given an unsorted array 10,16,8,12,15,6,3,9,5 Write a program to perform Quick Sort. Choose the first element as the pivot and partition the array accordingly. Show the array after this partition. Recursively apply Quick Sort on the sub-arrays formed. Display the array after each recursive call until the entire array is sorted.

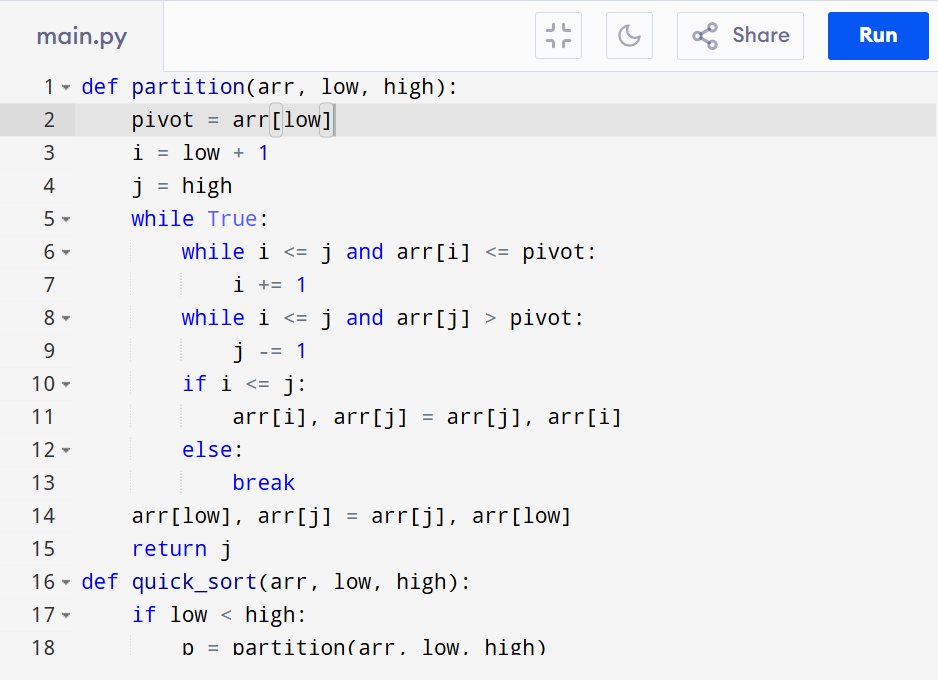
**AIM**

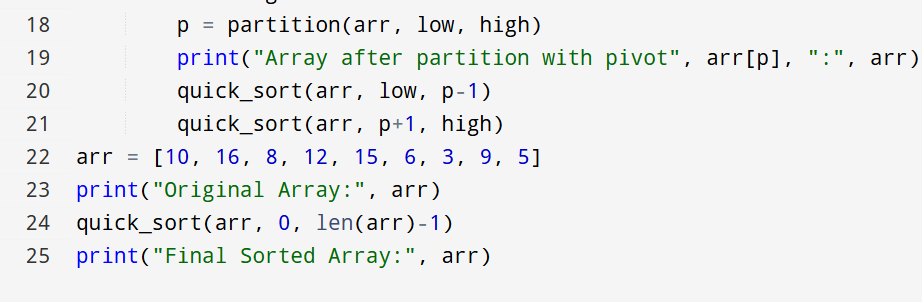
To implement **Quick Sort** for sorting an unsorted array, choosing the **first element** as the pivot and showing the array after each recursive step.

**ALGORITHM**

1. Input the array of size **N**.
2. If low < high:- Partition the array and Choose the first element as pivot.
3. Rearrange elements so that all elements smaller than pivot are on the left, and larger ones on the right.
4. Place pivot in its correct sorted position.
5. Recursively apply Quick Sort on left subarray.
6. Recursively apply Quick Sort on right subarray.
7. Display the array after each recursive call.

**PROGRAM**

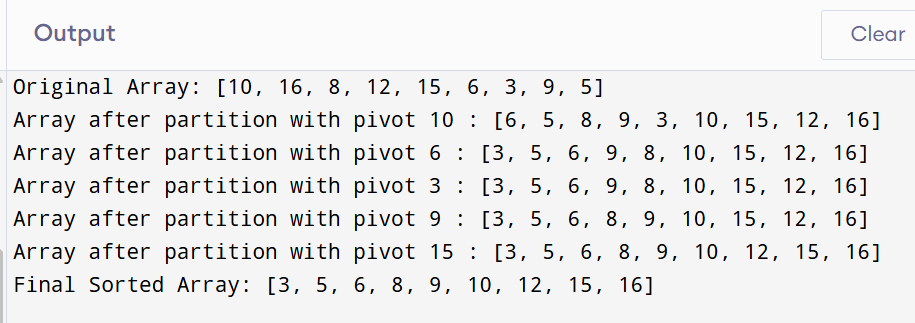




Input:

N=9, a[]= {10,16,8,12,15,6,3,9,5}

Output:



**RESULT:**

Thus the program to implement quick sort is successfully executed and the output is verified.

**PERFORMANCE ANALYSIS:**

* **Time Complexity**:
* Best & Average Case: **O(N log N)**
* Worst Case (already sorted, pivot chosen badly): **O(N²)**

· **Space Complexity**:

* · Recursive stack space: **O(log N)** (best), **O(N)** (worst)