160. 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.

Input : N= 9, a[]= {10,16,8,12,15,6,3,9,5}
Output : 3,5,6,8,9,10,12,15,16
Test Cases :
Input : N= 8, a[] = {12,4,78,23,45,67,89,1}
Output : 1,4,12,23,45,67,78,89
Test Cases :
Input : N= 7, a[] = {38,27,43,3,9,82,10}
Output : 3,9,10,27,38,43,82,

PROGRAM :-

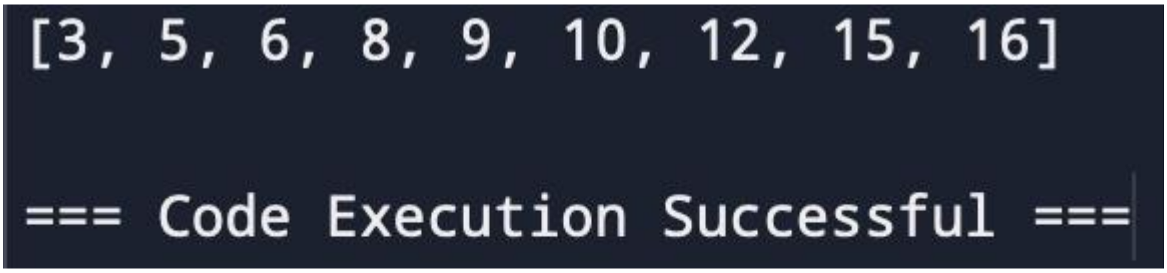
```
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        less_than_pivot = [x for x in arr[1:] if x <= pivot]
        greater_than_pivot = [x for x in arr[1:] if x > pivot]
        return quick_sort(less_than_pivot) + [pivot] + quick_sort(greater_than_pivot)

# Input array
arr = [10, 16, 8, 12, 15, 6, 3, 9, 5]

# Perform Quick Sort
sorted_arr = quick_sort(arr)

print(sorted_arr)
```

OUTPUT:-



```
[3, 5, 6, 8, 9, 10, 12, 15, 16]

=== Code Execution Successful ===
```

TIME COMPLEXITY:- O(N log N)