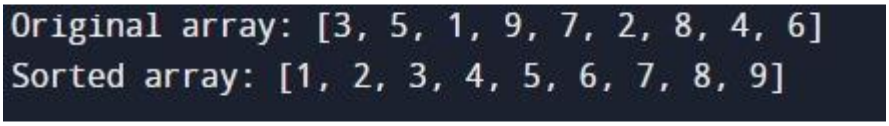
80.Merge Sort

AIM: To sort the elements with Merge Sort by using Divide and Conquer method

PROGRAM:

```
def merge_sort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left_half = arr[:mid]
    right_half = arr[mid:]
    left_sorted = merge_sort(left_half)
    right_sorted = merge_sort(right_half)
    sorted_arr = merge(left_sorted, right_sorted)
    return sorted_arr
def merge(left, right):
    merged = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            merged.append(left[i])
            i += 1
        else:
            merged.append(right[j])
            j += 1
    merged.extend(left[i:])
    merged.extend(right[j:])
    return merged
arr = [3, 5, 1, 9, 7, 2, 8, 4, 6]
sorted_arr = merge_sort(arr)
print(f"Original array: {arr}")
print(f"Sorted array: {sorted_arr}")
```

OUTPUT:



```
Original array: [3, 5, 1, 9, 7, 2, 8, 4, 6]
Sorted array: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

TIME COMPLEXITY: O (n log n)