3.3 MERGE SORT IMPLEMENTATION FOR UNSORTED ARRAY

Question:

You are given an unsorted array 54 28 03 15 12 7 3. Write a program for Merge Sort and implement using any programming language of your choice.

AIM

To implement the Merge Sort algorithm in Python and sort the given unsorted array of integers.

ALGORITHM

1. Divide the array into two halves recursively until each sub-array contains a single element.
2. Conquer by merging the sub-arrays in sorted order.
3. Repeat the merge process until the entire array is sorted.

PROGRAM

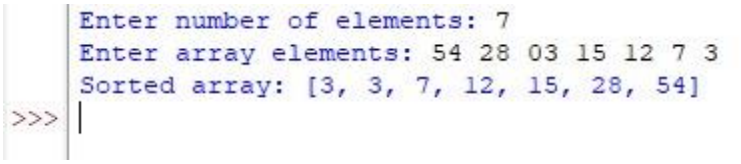
```
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = merge_sort(arr[:mid])
        R = merge_sort(arr[mid:])
        return merge(L, R)
    return arr

def merge(left, right):
    result = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    result.extend(left[i:])
    result.extend(right[j:])
    return result

def run_merge_sort():
    N = int(input("Enter number of elements: "))
    arr = list(map(int, input("Enter array elements: ").split()))
    print("Sorted array:", merge_sort(arr))

run_merge_sort()
```

Input:

7

54 28 03 15 12 7 3

Output:



RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: O(n log n), efficient for large datasets
- Space Complexity: O(n) due to auxiliary arrays used during merging