S.No: 4	Exp. Name: ***Merge Sort***	**Date:2025-09-24**

Aim:

Write a C program to perform Merge sort. Display the partial pass-wise sorting done.

Source Code:

mergeSortAlgo.c

```c
#include <stdio.h>
#include <stdlib.h>

void print_subarray(int a[], int l, int r) {
for (int i = l; i <= r; i++) {
printf("%d ", a[i]);
if (i < r)
printf("");
}
printf("\n");
}

void merge(int a[], int l, int m, int r, int temp[]) {
int i = l, j = m + 1, k = l;
while (i <= m && j <= r) {
if (a[i] <= a[j])
temp[k++] = a[i++];
else
temp[k++] = a[j++];
}
while (i <= m)
temp[k++] = a[i++];
while (j <= r)
temp[k++] = a[j++];
for (i = l; i <= r; i++)
a[i] = temp[i];
printf("Pass: ");
print_subarray(a, l, r);
}
void mergesort(int a[], int l, int r, int temp[]) {
if (l >= r)
return;
int m = (l + r) / 2;
mergesort(a, l, m, temp);
mergesort(a, m + 1, r, temp);
merge(a, l, m, r, temp);
}

int main() {
int n;
printf("no of elements: ");
scanf("%d", &n);

int *a = (int *)malloc(n * sizeof(int));
int *temp = (int *)malloc(n * sizeof(int));
printf("elements: ");
for (int i = 0; i < n; i++) {
scanf("%d", &a[i]);
}
printf("Given array:\n");
for (int i = 0; i < n; i++) {
printf("%d", a[i]);
if (i < n - 1)
printf(" ");
}
printf(" \n");

mergesort(a, 0, n - 1, temp);

printf("Sorted array:\n");
for (int i = 0; i < n; i++) {
printf("%d", a[i]);
if (i < n - 1)
printf(" ");
}
printf(" \n");

free(a);
free(temp);
return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
no of elements:  5
elements:  5 3 7 1 9
Given array:
5 3 7 1 9
Pass: 3 5
Pass: 3 5 7
Pass: 1 9
Pass: 1 3 5 7 9
Sorted array:
1 3 5 7 9
```

Test Case - 2

User Output

```
no of elements:  8
elements:  8 4 2 7 1 5 3 6
Given array:
8 4 2 7 1 5 3 6
Pass: 4 8
Pass: 2 7
```

```
Pass: 2 4 7 8
Pass: 1 5
Pass: 3 6
Pass: 1 3 5 6
Pass: 1 2 3 4 5 6 7 8
Sorted array:
1 2 3 4 5 6 7 8
```