

Write a c Program to implement Merge Sort

```
1. #include <stdio.h>
2.
3. /* Function to merge the subarrays of a[] */
4. void merge(int a[], int beg, int mid, int end)
5. {
6.     int i, j, k;
7.     int n1 = mid - beg + 1;
8.     int n2 = end - mid;
9.
10.    int LeftArray[n1], RightArray[n2]; //temporary arrays
11.
12.    /* copy data to temp arrays */
13.    for (int i = 0; i < n1; i++)
14.        LeftArray[i] = a[beg + i];
15.    for (int j = 0; j < n2; j++)
16.        RightArray[j] = a[mid + 1 + j];
17.
18.    i = 0; /* initial index of first sub-array */
19.    j = 0; /* initial index of second sub-array */
20.    k = beg; /* initial index of merged sub-array */
21.
22.    while (i < n1 && j < n2)
23.    {
24.        if(LeftArray[i] <= RightArray[j])
25.        {
26.            a[k] = LeftArray[i];
27.            i++;
28.        }
29.        else
30.        {
31.            a[k] = RightArray[j];
32.            j++;
33.        }
```

```
34.         k++;
35.     }
36.     while (i<n1)
37.     {
38.         a[k] = LeftArray[i];
39.         i++;
40.         k++;
41.     }
42.
43.     while (j<n2)
44.     {
45.         a[k] = RightArray[j];
46.         j++;
47.         k++;
48.     }
49. }
50.
51. void mergeSort(int a[], int beg, int end)
52. {
53.     if (beg < end)
54.     {
55.         int mid = (beg + end) / 2;
56.         mergeSort(a, beg, mid);
57.         mergeSort(a, mid + 1, end);
58.         merge(a, beg, mid, end);
59.     }
60. }
61.
62. /* Function to print the array */
63. void printArray(int a[], int n)
64. {
65.     int i;
66.     for (i = 0; i < n; i++)
67.         printf("%d ", a[i]);
```

```
68.     printf("\n");
69. }
70.
71. int main()
72. {
73.     int a[] = { 12, 31, 25, 8, 32, 17, 40, 42 };
74.     int n = sizeof(a) / sizeof(a[0]);
75.     printf("Before sorting array elements are - \n");
76.     printArray(a, n);
77.     mergeSort(a, 0, n - 1);
78.     printf("After sorting array elements are - \n");
79.     printArray(a, n);
80.     return 0;
    }
```