

## DSA\_College\mergeSort.cpp

```
1  #include<iostream>
2  using namespace std ;
3
4  void merge(int* arr, int l, int mid, int r)
5  {
6      int n1 = mid-l+1 ;
7      int n2 = r-mid ;
8
9      int* left = new int[n1] ;
10     int* right = new int[n2] ;
11
12     for(int i=0;i<n1;i++)
13     {
14         left[i] = arr[l+i] ;
15     }
16     for(int i=0;i<n2;i++)
17     {
18         right[i] = arr[mid+1+i] ;
19     }
20
21     int i=0 ;
22     int j=0 ;
23     int k=l ;
24
25     while(i<n1 && j<n2)
26     {
27         if(left[i] < right[j])
28         {
29             arr[k] = left[i] ;
30             i++ ;
31             k++ ;
32         }
33         else{
34             arr[k] = right[j] ;
35             j++ ;
36             k++ ;
37         }
38     }
39
40     while(i<n1)
41     {
42         arr[k] = left[i] ;
43         i++ ;
44         k++ ;
45     }
46
47     while(j<n2)
48     {
49         arr[k] = right[j] ;
50         j++ ;
51         k++ ;
```

```
52     }
53 }
54
55 void mergeSort(int arr[], int l, int r)
56 {
57     if(l >= r)
58         return ;
59
60     int mid = l+ (r-l)/2 ;
61     mergeSort(arr,l,mid) ;
62     mergeSort(arr,mid+1,r) ;
63     merge(arr,l,mid,r) ;
64 }
65
66 void print(int arr[], int n)
67 {
68     for(int i=0;i<n;i++)
69     {
70         cout<<arr[i]<<" ";
71     }
72     cout<<endl;
73 }
74
75 int main()
76 {
77     int arr[] = {24,55,12,9,100,40,32} ;
78     int n= sizeof(arr)/sizeof(arr[0]) ;
79     mergeSort(arr,0,n-1);
80     print(arr,n) ;
81
82     return 0;
83 }
```