S.No: 11	Exp. Name: ***Write a C program to Sort given elements using Merge sort***	Date:2023-05-07

Aim:

Write a program to `sort` (`Ascending order`) the given elements using `merge sort` technique.

At the time of execution, the program should print the message on the console as:

```
Enter array size :
```

For example, if the user gives the **input** as:

```
Enter array size : 5
```

Next, the program should print the following message on the console as:

```
Enter 5 elements :
```

if the user gives the **input** as:

```
Enter 5 elements : 34 67 12 45 22
```

then the program should **print** the result as:

```
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67
```

Note: Do use the **printf()** function with a **newline** character (`\n`).

Source Code:

MergeSortMain.c

```
#include<stdio.h>
void main()
{
int arr[15],i,n;
printf("Enter array size : ");
scanf("%d", &n);
printf("Enter %d elements : ",n);
for(i=0;i<n;i++)
{
scanf("%d", &arr[i]);
}
printf("Before sorting the elements are : ");
display(arr, n);
splitAndMerge(arr, 0, n - 1);
printf("After sorting the elements are : ");
display(arr, n);
}
void display(int arr[15], int n)
{
int i;
for(i=0;i<n;i++)
printf("%d ",arr[i]);
printf("\n");
}
void merge(int arr[15],int low, int mid, int high) {
```

```
int i=low,h=low,j=mid+1,k,temp[15];
while(h<=mid&&j<=high)
{
   if(arr[h]<=arr[j])
    {
       temp[i]=arr[h];
       h++;
       }
       else
       {
          temp[i]=arr[j];
        j++;
       }
       i++;
    }
    if(h>mid)
    {
      for(k=j;k<=high;k++)
      {
         temp[i]=arr[k];
         i++;
      }
    }
    else
    {
      for(k=h;k<=mid;k++)
      {
         temp[i] = arr[k];
         i++;
      }
    }
    for(k=low;k<=high;k++)
    {
      arr[k]=temp[k];
    }
}
void splitAndMerge(int arr[15],int low, int high)
{
 if(low<high)
 {
   int mid=(low+high)/2;
   splitAndMerge(arr,low,mid);
   splitAndMerge(arr,mid+1,high);
   merge(arr,low,mid,high);
 }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
Enter array size :  5
Enter 5 elements :  34 67 12 45 22
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67
```

Test Case - 2

User Output

```
Enter array size :  8
Enter 8 elements :  77 55 22 44 99 33 11 66
Before sorting the elements are : 77 55 22 44 99 33 11 66
After sorting the elements are : 11 22 33 44 55 66 77 99
```

Test Case - 3

User Output

```
Enter array size :  5
Enter 5 elements :  -32 -45 -67 -46 -14
Before sorting the elements are : -32 -45 -67 -46 -14
After sorting the elements are : -67 -46 -45 -32 -14
```