S.No: 6	Exp. Name: ***Sort a list using Merge Sort Technique.***	Date:2023-09-16

Aim:

Write a java program to sort the given list of elements using **Merge Sort**.

Source Code:

q36416/MergeSort.java

```
package q36416;
import java.util.*;
class MergeSort{
   public static void main(String args[]){
      int n,i;
      Scanner sc= new Scanner(System.in);
      System.out.print("Enter no of elements: ");
      n=sc.nextInt();
      int a[]=new int[n];
      System.out.println("Enter the elements:");
      for(i=0;i<n;i++)
      a[i]=sc.nextInt();
      Merge.SplitAndMerge(a,0,n-1,n);
      System.out.print("Sorted array: \n");
      for(i=0;i<n;i++)
      {
         System.out.print(a[i]+" ");
      }

   }
}
class MergeSortDemo{
   public static void MergeSorting(int a[],int low,int mid,int high,int n){
      int i,j,k;
      int b[]=new int [n];
      i=low;
      j=mid+1;
      k=low;
      while(i<=mid&&j<=high)
      {
         if(a[i]<=a[j])
         {
            b[k]=a[i];
            i++;
         }
         else
         {
            b[k]=a[j];
            j++;
         }
         k++;
      }
      if(i<=mid)
      {
         while(i<=mid)
         {
```


ID: 22K61A05H2

```
            b[k]=a[i];
            i++;
            k++;
         }
      }
      else
      {
         while(j<=high)
         {
            b[k]=a[j];
            j++;
            k++;
         }
      }
      for(k=low;k<=high;k++)
       a[k]=b[k];
   }
}
class Merge{
   public static void SplitAndMerge(int a[],int low,int high,int n){
      int mid;
      if(low<high)
      {
         mid=(low+high)/2;
         Merge.SplitAndMerge(a,low,mid,n);
         Merge.SplitAndMerge(a,mid+1,high,n);
         MergeSortDemo.MergeSorting(a,low,mid,high,n);
      }
   }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter no of elements: 3
Enter the elements: 100 50 75
Sorted array:
50 75 100

Test Case - 2
User Output
Enter no of elements: 4
Enter the elements: 1 3 5 2
Sorted array:
1 2 3 5