**1)  Sort a given set of N integer elements using the Merge Sort technique and compute its time taken. Run the program for different values of N and record the time taken to sort.**

#include<stdio.h>

int arr[20]; // array to be sorted

int main()

{

int n,i;

printf("Enter the size of array\n"); // input the elements

scanf("%d",&n);

printf("Enter the elements:");

for(i=0;i<n;i++)

scanf("%d",&arr[i]);

merge\_sort(arr,0,n-1); // sort the array

printf("Sorted array:"); // print sorted array

for(i=0;i<n;i++)

printf("%d\t",arr[i]);

return 0;

}

int merge\_sort(int arr[],int low,int high)

{

int mid;

if(low<high)

{

mid=(low+high)/2;

// Divide and Conquer

merge\_sort(arr,low,mid);

merge\_sort(arr,mid+1,high);

// Combine

merge(arr,low,mid,high);

}

return 0;

}

int merge(int arr[],int l,int m,int h)

{

int arr1[10],arr2[10]; // Two temporary arrays to

// hold the two arrays to be merged

int n1,n2,i,j,k;

n1=m-l+1;

n2=h-m;

for(i=0;i<n1;i++)

arr1[i]=arr[l+i];

for(j=0;j<n2;j++)

arr2[j]=arr[m+j+1];

arr1[i]=9999; // To mark the end of each temporary array

arr2[j]=9999;

i=0;j=0;

for(k=l;k<=h;k++) //process of combining two sorted arrays

{

if(arr1[i]<=arr2[j])

arr[k]=arr1[i++];

else

arr[k]=arr2[j++];

}

return 0;

}