

Ex. No:11.B.2**MERGE SORT****AIM:**

To write a C program to implement the concept of merge sort.

DESCRIPTION:

Merge sort is a sorting algorithm that uses the divide, conquer, and combine algorithmic paradigm.

Divide means partitioning the n-element array to be sorted into two sub-arrays of $n/2$ elements.

If there are more elements in the array, divide A into two sub-arrays, A1 and A2, each containing about half of the elements of A.

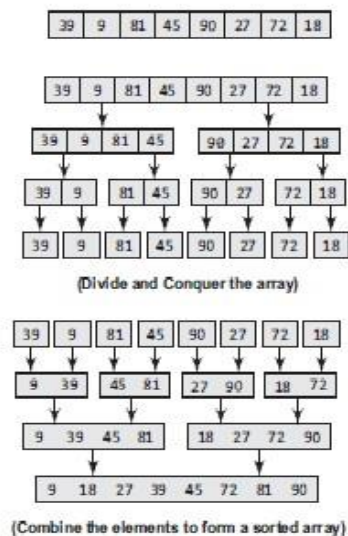
Conquer means sorting the two sub-arrays recursively using merge sort.

Combine means merging the two sorted sub-arrays of size $n/2$ to produce the sorted array of n elements.

The basic steps of a merge sort algorithm are as follows:

- a. If the array is of length 0 or 1, then it is already sorted.
- b . Otherwise, divide the unsorted array into two sub-arrays of about half the size. Use merge sort algorithm recursively to sort each sub-array.

Merge the two sub -arrays to form a single sorted list

**ALGORITHM:**

1: Start.

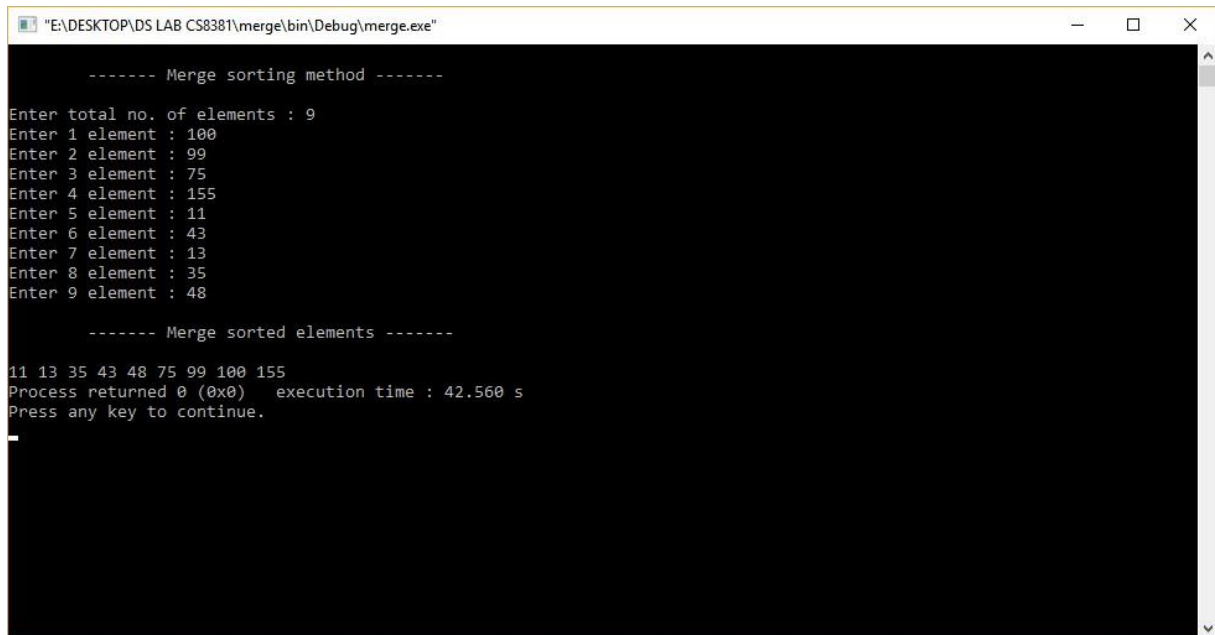
- 2: First you divide the number of elements by 2 and separate them as two.
- 3: Divide those two which are divided by 2.
- 4: Divide them until you get a single element.
- 5: Start comparing the starting two pair of elements with each other and place them in ascending order.
- 6: When you combine them compare them so that you make sure they are sorted.
- 7: When all the elements are compared the array will be surely sorted in an ascending order.
- 8: Stop.

PROGRAM:

```
#include<stdio.h> #include<conio.h> void merge(int
[],int ,int ,int ); void part(int [],int ,int );
void main(){ int arr[30]; int i,size;
printf("\n\t----- Merge sorting method -----
\n\n"); printf("Enter total no. of elements : ");
scanf("%d",&size); for(i=0; i<size; i++){
printf("Enter %d element : ",i+1);
scanf("%d",&arr[i]);
} part(arr,0,size-1); printf("\n\t----- Merge
sorted elements ----- \n\n"); for(i=0; i<size;
i++) printf("%d ",arr[i]); getch(); } void part(int
arr[],int min,int max){ int mid; if(min<max){
mid=(min+max)/2; part(arr,min,mid);
part(arr,mid+1,max); merge(arr,min,mid,max);}} void
merge(int arr[],int min,int mid,int max){ int
tmp[30];
int i,j,k,m; j=min; m=mid+1;
for(i=min; j<=mid && m<=max ;
i++){ if(arr[j]<=arr[m]){
tmp[i]=arr[j]; j++;} else{
```

```
tmp[i]=arr[m]; m++; }}  
if(j>mid){ for(k=m; k<=max;  
k++){ tmp[i]=arr[k]; i++; }}  
else{ for(k=j; k<=mid; k++){  
tmp[i]=arr[k]; i++; }}  
for(k=min; k<=max; k++)  
arr[k]=tmp[k]; }
```

OUTPUT



```
"E:\DESKTOP\DS LAB CS8381\merge\bin\Debug\merge.exe"  
----- Merge sorting method -----  
Enter total no. of elements : 9  
Enter 1 element : 100  
Enter 2 element : 99  
Enter 3 element : 75  
Enter 4 element : 155  
Enter 5 element : 11  
Enter 6 element : 43  
Enter 7 element : 13  
Enter 8 element : 35  
Enter 9 element : 48  
  
----- Merge sorted elements -----  
11 13 35 43 48 75 99 100 155  
Process returned 0 (0x0)   execution time : 42.560 s  
Press any key to continue.
```

RESULT:

Thus a C program for the concept of merge sort was implemented successfully.