**Experiment 2 :**

#include <iostream>

using namespace std;

void merge(int arr[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

int L[n1], R[n2];

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

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

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

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

i = 0; // Initial index of first subarray

j = 0; // Initial index of second subarray

k = l; // Initial index of merged subarray

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergeSort(int arr[], int l, int r)

{

if (l < r)

{

// Same as (l+r)/2, but avoids overflow for

// large l and h

int m = l+(r-l)/2;

// Sort first and second halves

mergeSort(arr, l, m);

mergeSort(arr, m+1, r);

merge(arr, l, m, r);

}

}

int main()

{

int arr[1000], n, i;

cout<<"Welcome To MERGE SORT"<<endl;

cout<<"Enter the no. of elements"<<endl;

cin>>n;

cout<<"Enter the value of elements"<<endl;

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

{

cin>>arr[i];

}

cout<<"Before sorting"<<endl;

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

{

cout<<":"<<arr[i]<<endl;

}

mergeSort(arr,0,n-1);

cout<<"After sorting"<<endl;

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

{

cout<<":"<<arr[i]<<endl;

}

return 0;

}

**OUTPUT**

Welcome To MERGE SORT

Enter the no. of elements

5

Enter the value of elements

98

34

675

-756

4

Before sorting

:98

:34

:675

:-756

:4

After sorting

:-756

:4

:34

:98

:675