**PROGRAM 8**

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

**CODE:**

#include<stdio.h>

#include<time.h>

#include<stdlib.h>

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);

}

}

void printArray(int A[], int size)

{

int i;

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

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

}

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

{

int i, j, k;

int p = m - l + 1;

int q = r - m;

int L[p], R[q];

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

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

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

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

i = 0;

j = 0;

k = l;

while (i < p && j < q)

{

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

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < p) {

arr[k] = L[i];

i++;

k++;

}

while (j < q) {

arr[k] = R[j];

j++;

k++;

}

}

int main()

{

int i,n,arr[1000];

double time;

clock\_t start,end;

printf("Enter the total number of elements to be sorted:");

scanf("%d",&n);

printf("The elements to be sorted are:\n");

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

{

arr[i] = rand()%100;

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

}

printf("\nThe sorted array is:\n");

start=clock();

mergeSort(arr, 0, n- 1);

printArray(arr,n-1);

end=clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("\nTime taken to sort the array by using Merge Sort is:%lf\n",time);

getch();

}

**OUTPUT:**

