**MERGE SORT**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

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;

j = 0;

k = l;

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)

{

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

mergeSort(arr, l, m);

mergeSort(arr, m+1, r);

merge(arr, l, m, r);

}

}

int main()

{

int arr[100000];

int n;

clock\_t start,end;

for(;;){

printf("Enter the number of elements:\n");

scanf("%d",&n);

printf("Enter elements to array:\n");

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

{

arr[i] = rand();

}

start = clock();

mergeSort(arr, 0, n-1);

end = clock();

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

printf("Sorted array is:\n");

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

{

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

}

printf("time required is %.8f\n",((double)(end-start)/CLOCKS\_PER\_SEC));

}}

