LAB-8

MERGE SORT CODE-

```
#include<stdlib.h>
#include<stdio.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);
    }
}

void printArray(int A[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

int main()
{
int i,n,sort;
clock_t start,end;
while(1)
{
printf("Enter the number of the elements\n");
scanf("%d",&n);
if(n==-1)
    break;
int a[n];
for(i=0;i<n;i++)
{
a[i]=rand();
}

start=clock();
mergeSort(a, 0, n - 1);

printf("Sorted array:\n");

   printArray(a, n);

end=clock();

double time_taken=((double)end-start)/CLOCKS_PER_SEC;

printf("\n\n");

printf("Time taken for sorting %d elements is %f sec\n",n,time_taken);

printf("\n");

}

}
```