Aim:

Write a C program to perform Merge sort. Display the partial pass-wise sorting done.

Source Code:

mergeSortAlgo.c

```c
// Type Content here...
#include<stdio.h>
int pass = 1;

void display(int a[], int start, int end) {
    for(int i=start; i<=end; i++){
        printf("%d ", a[i]);
    }
    printf("\n");
}
void merge(int a[], int l, int m, int r) {
    int n1 = m-l+1;
    int n2 = r-m;

    int L[100], R[100];

    for(int i=0; i<n1; i++)
        L[i] = a[l+i];
    for(int j=0; j<n2; j++)
        R[j] = a[m+1+j];

    int i=0, j=0, k=l;

    while(i<n1&&j<n2) {
        if(L[i] <= R[j]){
            a[k++] = L[i++];
        } else{
            a[k++] = R[j++];
        }
    }
    while(i<n1)
        a[k++] = L[i++];

    while(j<n2)
        a[k++] = R[j++];


    if(r - l >= 1) {
        printf("Pass: ");
        display(a,l,r);
    }
}

void mergeSort(int a[], int l, int r) {
    if(l<r) {
        int m=(l+r)/2;
```

```
        mergeSort(a,l,m);
        mergeSort(a,m+1,r);
        merge(a,l,m,r);
    }
}

int main(){
    int a[100], n;
    printf("no of elements: ");
    scanf("%d", &n);

    printf("elements: ");
    for(int i=0; i<n; i++){
        scanf("%d", &a[i]);
    }
    printf("Given array:\n");
    display(a, 0, n-1);

    mergeSort(a, 0, n-1);

    printf("Sorted array:\n");
    display(a,0,n-1);

    return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
no of elements: 5
elements: 5 3 7 1 9
Given array:
5 3 7 1 9
Pass: 3 5
Pass: 3 5 7
Pass: 1 9
Pass: 1 3 5 7 9
Sorted array:
1 3 5 7 9

Test Case - 2
User Output
no of elements: 8
elements: 8 4 2 7 1 5 3 6
Given array:
8 4 2 7 1 5 3 6
Pass: 4 8
Pass: 2 7
Pass: 2 4 7 8

Pass: 1 5
Pass: 3 6
Pass: 1 3 5 6
Pass: 1 2 3 4 5 6 7 8
Sorted array:
1 2 3 4 5 6 7 8