

main.c



Run

Output

Clear

```
1 #192210211
2 #include <stdio.h>
3 #include <stdlib.h>
4 void merge(int arr[], int l, int m, int r){
5     int i, j, k;
6     int n1 = m - l + 1;
7     int n2 = r - m;
8     int L[n1], R[n2];
9     for (i = 0; i < n1; i++)
10         L[i] = arr[l + i];
11     for (j = 0; j < n2; j++)
12         R[j] = arr[m + 1 + j];
13     i = 0;
14     j = 0;
15     k = l;
16     while (i < n1 && j < n2) {
17         if (L[i] <= R[j]) {
18             arr[k] = L[i];
19             i++; }
20         else {
21             arr[k] = R[j];
22             j++; }
23         k++; }
24     while (i < n1) {
25         arr[k] = L[i];
26         i++;
27         k++; }
28     while (j < n2) {
29         arr[k] = R[j];
30         j++; }
```

/tmp/dKzoRx1qjh.o



Given array is

12 11 13 5 6 7

Sorted array is

5 6 7 11 12 13

main.c

  [Run](#)

```
23     k++; }
24     while (i < n1) {
25         arr[k] = L[i];
26         i++;
27         k++; }
28     while (j < n2) {
29         arr[k] = R[j];
30         j++;
31         k++;}}
32 void mergeSort(int arr[], int l, int r){
33     if (l < r) {
34         int m = l + (r - l) / 2;
35         mergeSort(arr, l, m);
36         mergeSort(arr, m + 1, r);
37         merge(arr, l, m, r);}}
38 void printArray(int A[], int size){
39     int i;
40     for (i = 0; i < size; i++)
41         printf("%d ", A[i]);
42     printf("\n");}
43 int main(){
44     int arr[] = { 12, 11, 13, 5, 6, 7 };
45     int arr_size = sizeof(arr) / sizeof(arr[0]);
46     printf("Given array is \n");
47     printArray(arr, arr_size);
48     mergeSort(arr, 0, arr_size - 1);
49     printf("\nSorted array is \n");
50     printArray(arr, arr_size);
51     return 0;
52 }
```

[Clear](#)

```
/tmp/dKzoRx1qjh.o
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13
```