11/11/24, 1:42 AM Merge Sort.c

SEM 3\Exp9\Merge_Sort.c

```
#include <stdio.h>
2
 3
   // Function to merge two subarrays
   void merge(int arr[], int left, int mid, int right)
 4
 5
        int i, j, k;
 6
 7
        int n1 = mid - left + 1;
 8
        int n2 = right - mid;
 9
10
        // Create temporary arrays
        int L[n1], R[n2];
11
12
13
        // Copy data to temporary arrays
14
        for (i = 0; i < n1; i++)</pre>
            L[i] = arr[left + i];
15
        for (j = 0; j < n2; j++)
16
17
            R[j] = arr[mid + 1 + j];
18
19
        // Merge the temporary arrays
20
        i = 0;  // Initial index of first subarray
                  // Initial index of second subarray
21
        k = left; // Initial index of merged subarray
22
23
        while (i < n1 && j < n2)
24
        {
25
            if (L[i] <= R[j])</pre>
26
            {
27
                arr[k] = L[i];
28
                i++;
            }
29
30
            else
31
            {
32
                arr[k] = R[j];
33
                j++;
34
            }
35
            k++;
        }
36
37
38
        // Copy remaining elements of L[], if any
39
        while (i < n1)
        {
40
41
            arr[k] = L[i];
42
            i++;
43
            k++;
44
        }
45
        // Copy remaining elements of R[], if any
46
47
        while (j < n2)
48
        {
49
            arr[k] = R[j];
50
            j++;
51
            k++;
```

```
52
53
   }
54
   // Function to perform merge sort
55
   void mergeSort(int arr[], int left, int right)
56
57
   {
58
        if (left < right)</pre>
59
        {
            int mid = left + (right - left) / 2;
60
61
62
            // Sort first and second halves
63
            mergeSort(arr, left, mid);
            mergeSort(arr, mid + 1, right);
64
            merge(arr, left, mid, right);
65
66
        }
67
68
```