

Merge Sort Code In C++

```
#include <iostream>
using namespace std;
```

```
// Function to merge two halves
void merge(int arr[], int left, int mid, int right) {
    int n1 = mid - left + 1; // size of left subarray
    int n2 = right - mid;    // size of right subarray
```

```
    // Create temporary arrays
    int L[n1], R[n2];
```

```
    // Copy data to temporary arrays
    for (int i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];
```

```
    // Merge the two temporary arrays back into arr
    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
```

```
    // Copy remaining elements of L[] if any
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }
```

```
// Copy remaining elements of R[] if any
while (j < n2) {
    arr[k] = R[j];
    j++;
    k++;
}
}
```

```
// Function to implement Merge Sort
void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        // Find the middle point
        int mid = left + (right - left) / 2;

        // Recursively sort first and second halves
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);

        // Merge the sorted halves
        merge(arr, left, mid, right);
    }
}
```

```
// Function to print array
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}
```

```
// Main function
int main() {
    int arr[] = {38, 27, 43, 3, 9, 82, 10};
    int n = sizeof(arr) / sizeof(arr[0]);
```

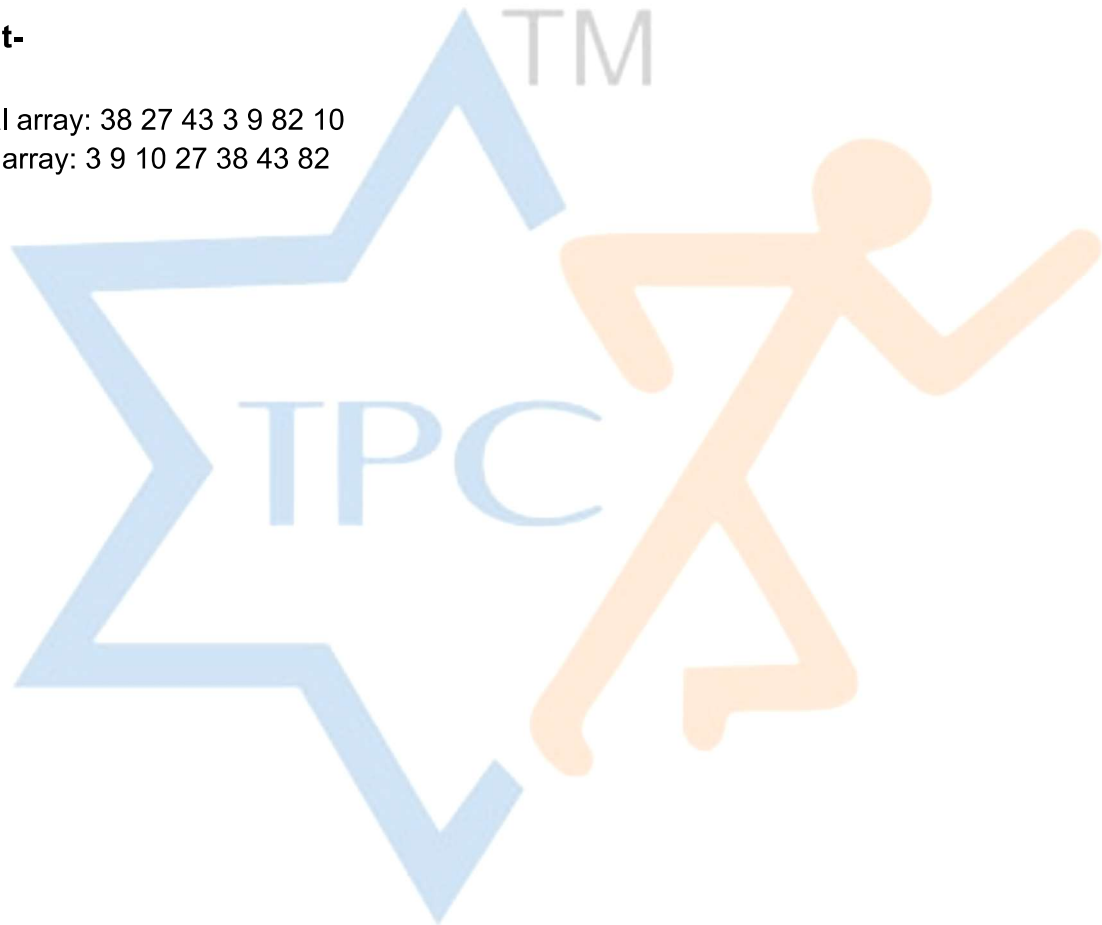
```
    cout << "Original array: ";
    printArray(arr, n);
```

```
    mergeSort(arr, 0, n - 1);
```

```
cout << "Sorted array: ";  
printArray(arr, n);  
  
return 0;  
}
```

Output-

Original array: 38 27 43 3 9 82 10
Sorted array: 3 9 10 27 38 43 82



www.tpcglobal.in