

```

#include <iostream>
using namespace std;
// Merge Function
void merge(int arr[], int left, int mid, int right)
{
    int i, j, k;

    int n1 = mid - left + 1;
    int n2 = right - mid;

    // Create temporary arrays
    int leftArray[n1], rightArray[n2];

    for (i = 0; i < n1; i++)
    {
        leftArray[i] = arr[left + i];
    }
    for (j = 0; j < n2; j++)
    {
        rightArray[j] = arr[mid + 1 + j];
    }

    i = j = 0;
    k = left;

    while (i < n1 && j < n2)
    {
        if (leftArray[i] <= rightArray[j])
        {
            arr[k] = leftArray[i];
            i++;
        }
        else
        {
            arr[k] = rightArray[j];
            j++;
        }
        k++;
    }

    while (i < n1)
    {
        arr[k] = leftArray[i];
    }
}

```

```

        i++, k++;
    }

    while (j < n2)
    {
        arr[k] = rightArray[j];
        j++, k++;
    }
}

// Recursive Merge Sort Function
void mergeSort(int arr[], int left, int right)
{
    if (left < right)
    {
        int mid = left + (right - left) / 2;

        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);

        merge(arr, left, mid, right);
    }
}

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

// Main Function
int main()
{
    int n;
    cout << "Enter number of elements: ";
    cin >> n;

    int arr[n];
    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

```

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

mergeSort(arr, 0, n - 1);

cout << "Sorted Array:  ";
printArray(arr, n);

return 0;
}
```