9) Merge sort

```
#include <iostream>
using namespace std;
class sort_tech
{
public:
   int arr[50], n;
   void get_Elements()
   {
      cout << "\n\tSORTING TECHNIQUES ";
      cout << "\n\t**********";
      cout << "\n Enter the size of Array:";
      cin >> n;
      cout << "\n Enter the Elements of array:";
      for (int i = 0; i < n; i++)
      {
         cin >> arr[i];
      }
   }
   void merge(int arr[], int left, int mid, int right)
   {
      int n1 = mid - left + 1;
      int n2 = right - mid;
      int l[n1], r[n2];
      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];
      }
      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++;
```

```
    }
    while (i < n1)
    {
        arr[k] = l[i];
        i++;
        k++;
    }
    while (j < n2)
    {
        arr[k] = r[j];
        j++;
        k++;
    }
}
void mergeSort(int arr[], int left, int right)
{
if (left >= right)
    {
        return;
    }
    int mid = left + (right - left) / 2;
    mergeSort(arr, left, mid);
    mergeSort(arr, mid + 1, right);
    merge(arr, left, mid, right); // This should be 'merge', not 'mergeSort'
}
void result()
{
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
}
};
int main()
{
    sort_tech st;
    st.get_elements();
    st.mergeSort(st.arr, 0, st.n - 1);
    st.result();
    return 0;
}
 if (left >= right)
    {
        return;
    }
    int mid = left + (right - left) / 2;
    mergeSort(arr, left, mid);
    mergeSort(arr, mid + 1, right);
```

```
    merge(arr, left, mid, right);
}
void result()
{
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
}
};
int main()
{
    sort_tech st;
    st.get_elements();
    st.mergeSort(st.arr, 0, st.n - 1);
    st.result();
    return 0;
}
```