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

// Two way Merge Sort
void merge(int arr[], int l, int r, int mid)
{
      int i, j, k, c[50];
      i = l;
      k = l;
      j = mid + 1;

      // Sorting the sub list and inserting the element into final list

      while (i <= mid && j <= r)
      {
            if (arr[i] < arr[j])
            {
                  c[k++] = arr[i++];
            }
            else
            {
                  c[k++] = arr[j++];
            }
      }

      // Assigning remaining elements of First Sub list

      while (i <= mid)
      {
            c[k++] = arr[i++];
      }

      // Assigning remaining elements of Second Sub list

      while (j <= r)
      {
            c[k++] = arr[j++];
      }

      // Assigning sorted elements into array

      for (i = l; i < k; i++)
      {
            arr[i] = c[i];
      }
}
// Merge Sort
void merge_sort(int arr[], int l, int r)
{
      int mid;
      if (l < r)
      {
            // find middle to divide the array into two halves

            mid = (l + r) / 2;

            // calling merge sort for first half

            merge_sort(arr, l, mid);

            // calling merge sort for second half

            merge_sort(arr, mid+1, r);
```

```
            // merge two sorted arrays
            merge(arr, l, r, mid);
      }
}

// main function
int main()
{
      int array[] = {9,3,7,5,6,4,8,2};
      int n;
      n = sizeof(array)/sizeof(array[0]);

      cout<<"Array element before Sorting : \n";
      for (int i = 0; i < n; i++)
            cout<<array[i]<<"\t";
      if (n == 0)
      {
            cout<<"\nThere is nothing to Sort";
            return 0;
      }
      else if (n == 1)
      {
            cout<<"\nSingle Element is already Sorted ";
      }
      else
      {
            merge_sort(array, 0, n);
            cout<<"\nArray elements after Sorting : \n";
            for (int i = 0; i < n; i++)
                  cout<<array[i]<<"\t";
      }
}
```