MERGE SORT :

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

void merge(int arr[], int l, int m, int r) {
  int i = l;
  int j = m + 1;
  int k = l;

  /* create temp array */
  int temp[5];

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

  }

  /* Copy the remaining elements of first half, if there are any */
  while (i <= m) {
    temp[k] = arr[i];
    i++;
    k++;

  }

  /* Copy the remaining elements of second half, if there are any */
  while (j <= r) {
    temp[k] = arr[j];
    j++;
    k++;
  }

  /* Copy the temp array to original array */
  for (int p = l; p <= r; p++) {
    arr[p] = temp[p];
  }
}

/* l is for left index and r is right index of the
   sub-array of arr to be sorted */
void mergeSort(int arr[], int l, int r) {
  if (l < r) {
    // find midpoint
    int m = (l + r) / 2;
    // recurcive mergesort first and second halves
    mergeSort(arr, l, m);
    mergeSort(arr, m + 1, r);

    // merge
    merge(arr, l, m, r);
  }
}

int main() {
  int myarray[5];
  //int arr_size = sizeof(myarray)/sizeof(myarray[0]);
  int arr_size = 5;

  cout << "Enter 5 integers in any order: " << endl;
  for (int i = 0; i < 5; i++) {
    cin >> myarray[i];
  }
  cout << "Before Sorting" << endl;
  for (int i = 0; i < 5; i++) {
    cout << myarray[i] << " ";
  }
  cout << endl;
  mergeSort(myarray, 0, (arr_size - 1)); //
mergesort(arr,left,right) called

  cout << "After Sorting" << endl;
  for (int i = 0; i < 5; i++) {
    cout << myarray[i] << " ";
  }

  return 0;
}
```