Merge sort is a sorting technique that sequences data by continuously merging items in a single sorted list. Every item in the original unordered list is merged with another, creating groups of two. Every two-item group is merged, creating groups of four and so on until there is one ordered list, as shown below.

50

80

70

15

20

10

45

90

65

20

10

45

90

65

20

10

45

90

65

20

10

20

10

20

10

90

45

65

45

65

45

65

90

45

65

90

20

10

45

65

90

70

15

50

80

70

15

50

80

70

15

50

80

70

15

50

80

15

50

80

70

70

15

65

50

70

80

90

45

20

10

15

1. Divide

The algorithm is basically as follows:

1. Divide the list in halves
2. Merge sort the first half
3. Merge sort the second half
4. Merge both halves back together

This algorithm lends itself well to recursion. The algorithm divides the array to be sorted into halves, sort these two sub-arrays separately, and then combine these sorted sub-arrays to produce solution to the original problem. Once the array is divided, the left sub-array is further divided into sub-arrays until the last sub-array has at most two values. At this point these two values are sorted; likewise, the sub-array to the neighbor (if any) this sorted array, is now sorted and is merged to the already sorted sub-list. Once the left sub-array is sorted and merged, the right sub-array is divided and sorted like the left. When both sub-arrays sorted, they are then merged.

merge\_sort( int arr[], int left\_index, int right\_index)

{

if (right\_index > left\_index)

{

int mid = (left\_index + right\_index)/2;

**int left\_sublist[ ]** = merge\_sort( arr, left\_index, mid);

**int right\_sublist[ ]** = merge\_sort( arr, mid+ 1, right\_index,);

merge(arr, left\_sublist, right\_sublist);

}

}

Write a program that implements the merge-sort algorithm