Sorting:

Sort the list shown below in ascending order using:

1. Bubblesort.
2. Insertion sort
3. Selection sort
4. Merge sort

34 10 18 5 85 12 98 2

You are required to show each passes numbers.

1. Pass 1, i=0: 10 18 5 34 12 85 2 98

Pass 2, i=1: 10 5 18 12 34 2 85 98

Pass 3, i=2: 5 10 12 18 2 34 85 98

Pass 4, i=3: 5 10 12 2 18 34 85 98

Pass 5, i=4: 5 10 2 12 18 34 85 98

Pass 6, i=5: 5 2 10 12 18 34 85 98

Pass 7, i=6: 2 5 10 12 18 34 85 98

Pass 8, i=7: 2 5 10 12 18 34 85 98

1. Pass 1, i=0: 10 34 18 5 85 12 98 2

Pass 2, i=1: 10 18 34 5 85 12 98 2

Pass 3, i=2: 5 10 18 34 85 12 98 2

Pass 4, i=3: 5 10 18 34 85 12 98 2

Pass 5, i=4: 5 10 12 18 34 85 98 2

Pass 6, i=5: 2 5 10 12 18 34 85 98

Pass 7, i=6: 2 5 10 12 18 34 85 98

1. Pass 1, i=0: 2 10 18 5 85 12 98 34

Pass 2, i=1: 2 5 18 10 85 12 98 34

Pass 3, i=2: 2 5 10 18 85 12 98 34

Pass 4, i=3: 2 5 10 12 85 18 98 34

Pass 5, i=4: 2 5 10 12 18 85 98 34

Pass 6, i=5: 2 5 10 12 18 34 98 85

Pass 7, i=6: 2 5 10 12 18 34 85 98

**Array division**

1. Find midpoint (left end index of array + right end index of array)/2

Example: 34 10 18 (5) 85 12 98 2, midpoint = (0+7)/2 = 3

1. Divide into left and right subarrays

Iteration 1: 34,10,18,5 (left subarray) | 85,12,98,2 (right subarray)

1. Repeat step 1 and 2 until subarray divided into individual blocks

Iteration 2: 34,10 | 18,5 | 85,12 | 98,2

Iteration 3: 34 | 10 | 18 | 5 | 85 | 12 | 98 | 2

**Conquering subarrays**

Compare value of previously related subarray blocks, starting from left subarrays, change value to smaller index if smaller value

From iteration 3 output, iteration for conquering subarrays:

Iteration 1: 10,34 | 5,18 | 85|12 | 98|2

Iteration 2 : 5,18,10,34 | 85|12| 98|2

Finished conquering left subarrays, moving on to right subarrays:  
Iteration 3: 5,18,10,34 | 12,85| 2,98

Iteration 4 : 5,18,10,34 | 2,12,85,98

Finished conquering right subarrays, join the 2 subarrays back to form the updated base array

Iteration 5: 2,5,10,12,18,34,85,98