```
// CMPT135 J.Ye
// Sorting an array in ascending order using Merge Sort algorithm

// Note: this version does not contain the implementation of the merge function, please
// try it on your own first.


void mergeSort(int arr[], int start, int end);
//Precondition: The array elements arr[start] through arr[end] have values. start <= end
//Postcondition: The values of arr[start] through arr[end] have been rearranged so that
//                          arr[start] <= arr[start+1] <= ... <= arr[end].


void merge(int arr[], int start, int mid, int end);
//Precondition: The array elements arr[start] through arr[end] have values. start <= mid <= end
//            subarray arr[start] to arr[mid] is sorted; subarray arr[mid+1] to arr[end] is sorted
//Postcondition: the entire array arr[start] to arr[end] is sorted



void mergeSort(int arr[], int start, int end)
{
       if (start == end-1)  // size 2 array, base case.  What about size 1 array?
       {
              // swap the two elements if they are out of order
              if(arr[start] > arr[end])
              {
                     int temp = arr[start];
                     arr[start] = arr[end];
                     arr[end] = temp;
              }
       }

       else if (start < end - 1) // size > 2 array
       {
              int mid = (start+end)/2;          // "cut" the array in half

              mergeSort(arr, start, mid);       // sort the first half

              mergeSort(arr, mid+1, end);       // sort the 2nd half

              merge(arr, start, mid, end);      // merge them back together
       }
}


// Now, please implement the merge function ...
```