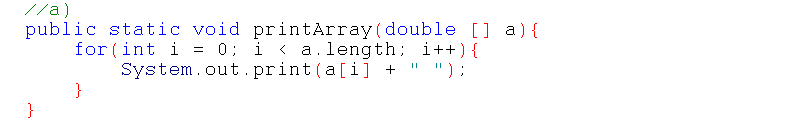
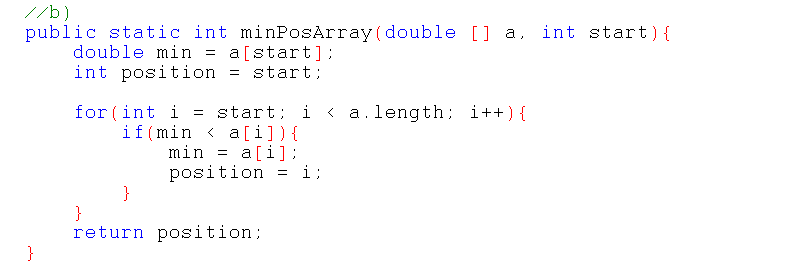
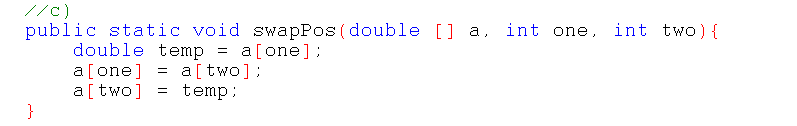
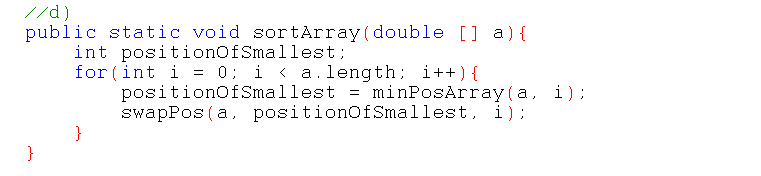
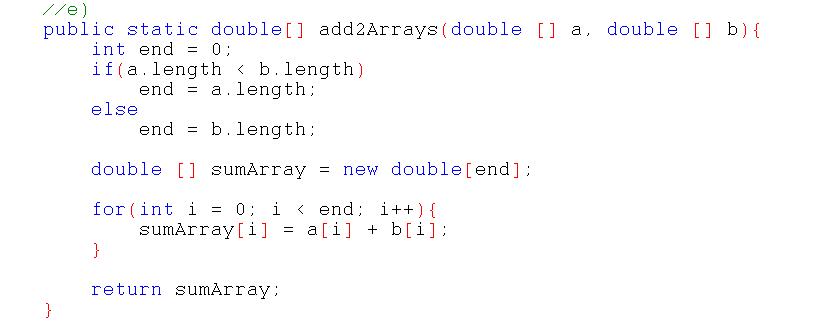
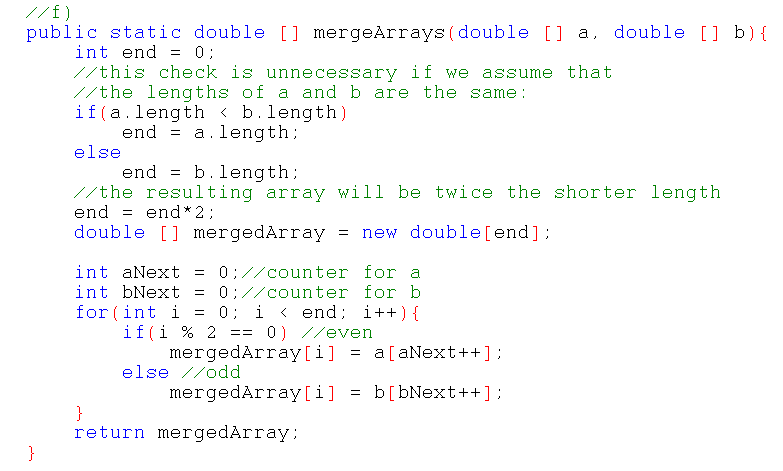
**Array Exercises - Sample Solutions**

a) Write a method printArray that, given an array of doubles, prints it.  
  
  
  
b) Write a method minPosArray that, given an array of doubles and a start position, returns the position in the rest of the array that contains the smallest value.  
  
  
  
c) Write a method swapPos that, given an Array of doubles and two positions in that array, swaps the values at those positions.  (No for loop required here.)  
  
  
  
d) Write a method sortArray that, given an array of doubles, will loop through the array and sort it: for each ith position from 1 to length of the array it will use minPosArray and swapPos to sort the smallest value into position i.  
  
  
  
e) Write a method called add2Arrays that, given two arrays of doubles, returns a third array where each position is the sum of the values at the same positions in the input arrays.  Assume both arrays are the same length.  Later discuss what would be done if one is shorter.  
  
  
  
f) Write a method called mergeArrays which given two arrays of doubles (assume they are exactly the same size) declares, populates and returns a single array of twice the size where the zeroth, 2nd 4th ... end-1 positions of the new array have the first input array's values, and the 1st, 3rd, 5th ... end positions have the second input arrays values.  
  
  
  
g) (Challenge) Write a method mergeSortedArrays which, given two sorted arrays (a and b), returns a single sorted array.  Duplicates are allowed and preserved since a separate method could be written to remove duplicates.  A possible solution is that the resulting array would have at it's current position the smaller of the next of a or next of b.  
