/\*\*

An insertion sort algorithm that uses recursion to

sort an array from low to high numbers.

@author Pj

\*\*/

import java.util.Random;

public class InsertionRecursion

{

public static void main(String[] args)

{

int[] array = new int[10];

Random randomObj = new Random();

//fills each index of the array with random numbers.

for (int i = 0; i < array.length; i++)

{

array[i] = randomObj.nextInt(1000);

System.out.print(array[i] + " ");

}

System.out.println();

Sort(array, 1); //calls the recursive sorting method and starts at index 1.

//prints out the sorted array

for (int j = 0; j < array.length; j++)

{

System.out.print(array[j] + " ");

}

}

public static void Sort(int[] array, int k)

{

if (k == array.length - 1) //Base case, if scanning at the end of array, sort and don't call insertionSort() again.

{

checks(array, k);

}

else //Recursive case, if sorting array is not done

{

checks(array, k);

k++;

Sort(array, k);

}

}

/\*\*

A method that sorts array by swapping the current index with the previous, if it is smaller.

insertionSort() calls this method to perform checks and swaps onto the array.

\*\*/

public static int[] checks(int[] array, int a)

{

if (a == 0) //Base case, if done with scanning the array.

{

return array;

}

if (array[a] < array[a-1]) /iIf previous index is larger, swap with it.

{

int temp = array[a-1];

array[a-1] = array[a];

array[a] = temp;

a--;

checks(array, a);

}

return array;

}

}