/\*\*

A radix sort algorithm that uses recursion to

sort an array from low to high numbers.

@author Pj Kim

\*\*/

import java.util.Random;

public class RadixRecursion

{

public static void main(String[] args)

{

int[] array = new int[10];

int[] temp = new int[array.length];

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();

radixSort(array, temp); //sorts the array using radix sort.

//prints out the sorted array.

for (int d = 0; d < temp.length; d++)

{

System.out.print(temp[d] + " ");

}

}

//sorts the array using radix sort.

public static void radixSort(int[] array, int[] temp)

{

sortFirst(array, temp, 0, 0, 0); //sorts 1's place

sortSecond(array, temp, 0, 0, 0); //sorts 10's place

sortThird(array, temp, 0, 0, 0); //sorts 100's place

}

//sorts the items considering the 1's place based on the temporary array and stores it in the temporary array

public static void sortFirst(int[] array, int[] temp, int j, int i, int a)

{

if (array[i]%10 == a)

{

temp[j] = array[i];

j++;

}

i++;

if (i == array.length)

{

i = 0;

a++;

}

if (a != 10)

{

sortFirst(array, temp, j, i, a);

}

}

//sorts the items considering 10's places based on the temporary array and stores it back in the array

public static void sortSecond(int[] array, int[] temp, int p, int i, int a)

{

if (((temp[i]%100)/10) == a)

{

array[p] = temp[i];

p++;

}

i++;

if (i == array.length)

{

i = 0;

a++;

}

if (a != 10)

{

sortSecond(array, temp, p, i, a);

}

}

// sorts the items considering 100's places based on the temporary array and stores it back in the array

public static void sortThird(int[] array, int[] temp, int x, int i, int a)

{

if ((array[i]/100) == a)

{

temp[x] = array[i];

x++;

}

i++;

if (i == array.length)

{

i = 0;

a++;

}

if (a != 10)

{

sortThird(array, temp, x, i, a);

}

}

}