/\*\*

A shell sort algorithm that uses iteration to

sort an array from low to high numbers.

@author Pj Kim

\*\*/

import java.util.Random;

public class ShellIteration

{

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();

//unscrambles values to make the insertion sort faster after

int hibbard = 3;

for (int j = 0; j < hibbard; j++)

{

int numOfComparisons = (array.length/hibbard) + 1;

if ((((numOfComparisons-1)\*hibbard) + j) > array.length-1)

{

numOfComparisons--;

}

for (int k = 1; k < numOfComparisons; k++)

{

for (int a = k; a > 0; a--)

{

if (array[(a\*hibbard)+j] < array[((w-1)\*hibbard)+j]) //if index is less than previous index, swap them.

{

int temp = array[((a-1)\*hibbard)+j];

array[((a-1)\*hibbard)+j] = array[(a\*hibbard)+j];

array[(a\*hibbard)+j] = temp;

}

else //If it's greater than or equal to the index in front,

{ //sort up to that index and come out of second for loop.

break;

}

}

}

}

//performs insertion sort

for (int b = 1; b < array.length; b++)

{

for (int c = b; c > 0; c--)

{

if (array[c] < array[c-1]) //If the index is less than the index in front, swap them.

{

int temp = array[c-1];

array[c-1] = array[c];

array[c] = temp;

}

else //if index is greater than or equal to the index in front,

{ //sort up to that index and come out of second for loop.

break;

}

}

}

//prints out the sorted array

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

{

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

}

}

}