public class HelloWorld{

private static void InsertSort(int[] a){

for(int i=1; i<a.length; i++){

if(a[i] < a[i-1]){

int x = a[i], j = i-1;

a[i] = a[i-1];

for(; j>=0 && x < a[j]; j--){

a[j+1] = a[j];

}

a[j+1] = x;

}

}

}

private static void ShellSort(int[] a){

for(int d=a.length/2; d>=1; d/=2){

for(int i=d; i<a.length; i++){

if(a[i] < a[i-d]){

int x = a[i], j = i - d;

a[i] = a[i-d];

for(; j>=0 && x < a[j]; j-=d){

a[j+d] = a[j];

}

a[j+d] = x;

}

}

}

}

private static void SelectSort(int[] a){

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

int k = i;

for(int j=i+1; j<a.length; j++){

if(a[k] < a[j])

k = j;

}

swap(a, i, k);

}

}

private static void swap(int[] a, int i, int j){

if(i == j) return;

int tmp = a[i];

a[i] = a[j];

a[j] = tmp;

}

private static void BubbleSort(int[] a, int n){

for(int i =0; i<n-1;i++){

for(int j=0; j<n-i-1;j++){

if(a[j] > a[j+1]){

int tmp = a[j];

a[j] = a[j+1];

a[j+1] = tmp;

}

}

}

}

private static void BubbleSort\_1(int[] a, int n){

int pos = n;

while(pos != 0){

int bound = pos;

pos = 0;

for(int i=0; i<bound - 1;i++){

if(a[i] > a[i+1]){

pos = i;

int tmp = a[i];

a[i] = a[i+1];

a[i+1] = tmp;

}

}

}

}

private static void BubbleSort\_2(int[] a, int n){

int low = 0, high = n - 1;

int i, tmp;

while(low < high){

for(i=low; i<high;i++){

if(a[i] > a[i+1]){

tmp = a[i];a[i] = a[i+1];a[i+1]=tmp;

}

}

high--;

for(i=high; i>low; i--){

if(a[i] < a[i-1]){

tmp = a[i-1];a[i-1]=a[i];a[i-1]=tmp;

}

}

low++;

}

}

private static void QuickSort(int[] a, int low, int high){

if(low < high){

int middle = getMiddle(a, low, high);

QuickSort(a, low, middle - 1);

QuickSort(a, middle + 1, high);

}

}

private static int getMiddle(int[] a, int low, int high){

int key = a[low];

while(low < high){

while((low < high) && (a[high] >= key)){

high--;

}

a[low] = a[high];

while((low < high) && (a[low] <= key)){

low++;

}

a[high] = a[low];

}

a[low] = key;

return low;

}

private static void MergeSort(int[] a, int[] a1, int n){

int h = 1;

while(h<n){

MergePass(a, a1, n, h);

h = 2\*h;

for(int i=0;i<n;i++)

a[i] = a1[i];

}

}

private static void MergePass(int[] a, int[] a1, int n, int h){

int i = 0;

while(i <= n - 2\*h){

Merge(a, a1, i, i+h-1, i+2\*h-1);

i += 2\*h;

}

if(i < n-h)

Merge(a, a1, i, i+h-1, n-1);

else

for(;i<n;i++)

a1[i] = a[i];

Print(a1);

}

private static void Merge(int[] a, int[] a1, int s, int m, int t){

int i = s, j = m + 1, k = s;

while(i <= m && j <= t){

if(a[i] < a[j])

a1[k++] = a[i++];

else

a1[k++] = a[j++];

}

if(i <= m)

while(i<=m) a1[k++] = a[i++];

if(j <= t)

while(j<=t) a1[k++] = a[j++];

}

private static void MergeSort(int[] a, int[] a1, int s, int e){

int[] a2 = new int[arr.length];

if(s == e) a1[s] = a[s];

else{

int m = (s + e)/2;

MergeSort(a, a2, s, m);

MergeSort(a, a2, m+1, e);

Merge(a2, a1, s, m, e);

}

}

private static void RadixSort(int[] a, int d){

int[][] array = new int[10][a.length+1];

for(int i=0; i < 10; i++){

array[i][0] = 0;

}

for(int i=1; i<=d; i++){

for(int j=0; j < a.length; j++){ // allocate

int row = getNumInPos(a[j], i);

int col = ++array[row][0];

array[row][col] = a[j];

}

for(int row=0, k = 0; row < 10; row++){ // collect

for(int col=1; col<=array[row][0]; col++){

a[k++] = array[row][col];

}

array[row][0] = 0; // reset for the next pos

}

}

}

private static int getNumInPos(int num, int pos){

int tmp = 1;

for(int i=0; i < pos - 1; i++){

tmp \*= 10;

}

return (num/tmp)%10;

}

private static int getMaxPos(int[] a){

int max = a[0];

// Find the max number in the Array

for(int i=1; i<a.length; i++){

if(a[i] > max)

max =a[i];

}

int res = 1;

while(max/10 !=0){

res++;

max /= 10;

}

return res;

}

private static int[] arr = { 49, 38, 65, 197, 76, 213, 27, 50 };

public static void main(String []args){

Print(arr);

//System.out.println("InsertSort");

//HelloWorld.InsertSort(arr);

//System.out.println("ShellSort");

//HelloWorld.ShellSort(arr);

//System.out.println("SelectSort");

//HelloWorld.SelectSort(arr);

//System.out.println("BubbleSort");

//HelloWorld.BubbleSort(arr, arr.length);

//System.out.println("BubbleSort\_1");

//HelloWorld.BubbleSort\_1(arr, arr.length);

//System.out.println("BubbleSort\_2");

//HelloWorld.BubbleSort\_2(arr, arr.length);

//System.out.println("QuickSort");

//HelloWorld.QuickSort(arr, 0, arr.length-1);

//System.out.println("MergeSort");

//int[] a1 = new int[arr.length];

//HelloWorld.MergeSort(arr, a1, arr.length);

//System.out.println("MergeSort");

//HelloWorld.MergeSort(arr, arr, 0, arr.length - 1);

System.out.println("RadixSort");

HelloWorld.RadixSort(arr, getMaxPos(arr));

Print(arr);

}

public static void Print(int[] a){

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

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

}

System.out.println("");

}

}