package com.company;

//Problem 2 assignment 2

//Time and space complexity – O(k) as radix sort is used

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Scanner;

public class Main {

public static void main(String[] args) throws IOException {

Scanner scan = new Scanner(System.in);

int k = scan.nextInt();

double max = (k\*k -1); //max upperlimit on input allowed – max range

int max\_len = 0,p = 0;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String[] s = new String[k];

s = br.readLine().split(" "); //Reading input

int[] nos = new int[k];

int[] res = new int[k];

int[] bucket = new int[10];

int digit;

for (int i = 0; i < k; i++) { //Stroing input in nos arrray

nos[i] = Integer.parseInt(s[i]);

}

for (int i = 0; i < 10; i++) { //Initializing bucket values to 0 for redix sort

bucket[i] = 0;

}

for (int i = 0; i < k; i++) { //Max no given by user

if(nos[i]>p){

p = nos[i];

}

}

max\_len = (int) Math.log(max);

max\_len++;

//redix sort from lsb to msb

for (int i = 1; p/i>0; i = i\*10) {

//count sort to sort individual position

for (int j = 0; j < k; j++) {

digit = (nos[j]/i)%10;

bucket[digit]++;

}

for (int j = 1; j < k; j++) {

bucket[j] = bucket[j]+bucket[j-1];

}

for (int j = k-1; j >= 0; j--) {

digit = (nos[j]/i)%10;

int l = bucket[digit] - 1;

res[l] = nos[j];

}

for (int j = 0; j < k; j++) {

nos[j] = res[j];

}

}

//nos[k-1]=p;

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

System.out.print(nos[i]+" "); //Output of radix sorting

}

}

}