Radix Sort

Problem Code	ds54b_quiz2_radix
Running Time Limit	1 sec
Memory Limit	128 mb

Introduction

A sorting problem rearrange data in an array **A** such that items in **A** is sorted in ascending order (**A[0]** is the smallest while **A[n-1]** is the largest). In this problem you have to write a Radix sort using Queue data structures.

We let digit(x,i) be the i^{th} digit of x where the right most digit is the 0^{th} digit. For example, digit(4268,0) is 8 while digit(4268,1) is 8. Radix sort works as follows.

1. We start with 10 empty Queues: q[0], q[1], …, q[9]. Let d be equal to 0.
2. Move items from A to the queues. Items in the array A is processed from A[0] to A[n-1]. For each A[i], it is enqueued into q[digit(A[i],d)].
3. Move items from the queues back to A. We start with q[0], dequeue all items from q[0] back to A. Repeat the same process on q[1], q[2], … q[9]. After this step, the queues are empty.
4. If d is still less than the number of digits in the largest number, increase d by one and go back to step 2, else everything is done.

Task

Write a radix sort function. The function is defined as void radix(int *A,int n). **DO NOT modify anything in the main function.**

You must write radix sort. **Any other sorting is not allowed**. However, the grader system will give you the score even you do not write a radix sort. Your code will be re-checked again by human. If the code is not the radix sort, the score will be reduced.

Input

The first line of input contains an integer n, indicating the number of items. The next line consists of n integers describing the array A. A[i] is at most 999,999,999. This means that the maximum value of d is 8.

Output

A single line containing sorted items of A.

Example

Ex1

Input	Output
4 4 3 2 1	1 2 3 4

Ex2

Input	Output
4 123 53 51 12	12 51 53 123