

4006_长度字母排序

描述

编写一个程序，输入一个句子，然后去掉其中所有重复出现的单词（只保留 1 个），并将剩余的单词按照“长度-字母”顺序进行排列，然后打印出来。所谓“长度-字母”顺序，就是把较短的单词放在前面，较长的单词放在后面，如果两个单词的长度相同，那么再按照字母顺序进行排序。

说明：（1）由于句子当中包含有空格，所以应该用 `gets` 函数来输入这个句子；（2）输入的句子当中只包含英文字符和空格，单词之间用一个空格隔开；（3）不用考虑单词的大小写，假设输入的都是小写字符；（4）句子长度不超过 100 个字符。

输入

输入只有一行，即一个英文句子。

输出

输出只有一行，即经过去重、排序之后得到的句子。

输入样例

```
jingle bells jingle bells jingle all the way
```

输出样例

```
all the way bells jingle
```

3052_敏感词检测

描述

作为一名网络警察，你的任务是监视电子邮件，看其中是否有一些敏感的关键词。

不过，有些狡猾的犯罪嫌疑人会改变某些单词的字母顺序，以逃避检查。

请编写一个程序，发现这种调整过顺序的关键词。

输入

程序的输入有两行，第一行是关键词列表，第二行是待检查的句子。

输出

程序的输出为在该句子中，所找到的关键词对应的原关键词，仅有一行，有多个关键词时，按照关键词在句子中的顺序输出，关键词之间用一个空格分隔； 如果一个关键词多次出现，则多次输出。

输入样例

```
guns mines missiles
```

```
aaameric sshell snug dan iimssle ot sit neemis
```

输出样例

```
guns missiles
```

3022_各数位出现次数

描述

编写一个程序，统计出在整数 N 的阶乘中，每一个十进制数据位（0~9）的出现次数。 N 的阶乘（ $N!$ ）指的是从 1 到 N 这 N 个整数的乘积，即 $N! = 1 \times 2 \times 3 \times \dots \times N$ 。阶乘的增长速度非常快，比如说， $13!$ 就已经比较大了，已经没法存放在一个整型变量当中；而 $35!$ 就更大了，它已经没有办法存放在一个浮点型变量当中。所以，当 N 比较大的时候，去计算 $N!$ 是非常困难的，无法直接用普通的数据变量来存储。例如， $10! = 3628800$ ，在这个整数中，0 出现了 2 次，2 出现了 1 次，3 出现了 1 次，6 出现了 1 次，8 出现了 2 次。而其余的数字的出现次数均为 0。

输入

输入只有一行，即一个整数 N （ $1 \leq N \leq 366$ ）。注意： $366!$ 是一个 781 位的巨大整数。

输出

输出 10 个整数，分别表示 0~9 在 $N!$ 中出现的次数。

输入样例

```
10
```

输出样例

```
2 0 1 1 0 0 1 0 2 0
```

3013_两组数据合并

描述

编写一个程序，对输入的两组数据进行合并。这两组数据均按照从小到大的顺序排列，且每组数据内部没有重复的元素。要求在合并以后，仍然保持从小到大的顺序排列，并且去掉可能出现的重复的元素。

说明：每组数据最少有 1 个，最多不超过 100 个。其中，每个数据都是一个整数，可能是正数或负数。每组数据内部没有重复的元素，但两组数据之间可能有重复的元素。

输入

输入有三行。第一行是 2 个正整数 **M** 和 **N**，分别表示两组数据的元素个数。第二行有 **M** 个整数，表示第一组数据。第三行有 **N** 个整数，表示第二组数据。

输出

输出只有一行，包括若干个整数，即合并以后的结果。相邻的两个整数之间用一个空格隔开。

输入样例

```
4 6
1 3 7 8
1 3 4 5 6 8
```

输出样例

```
1 3 4 5 6 7 8
```