

字符串排序

对字符串进行排序。

- 当字符串中包含数字时，比较数字大小。
 - 例如 `abc123` 和 `456ABC` 比较大小，由于两个字符串都包含数字，所以比较数字大小， $456 > 123$ ，因此 `456ABC > abc123`。
- 数字的大小比较是数值的大小比较，尽管它们出现在字符串中。
 - 例如 `A10` 和 `A2` 比较大小，由于 $10 > 2$ ，所以 `A10 > A2`。尽管在字符串的比较大中，字符 `A` 相等的前提下，字符 `2` > 字符 `1`，因此 `A2 > A10`，`strcmp` 不会继续进行 `0` 和 `\0` 的比较。
- 数字大小相同或不含数字时，沿用 `strcmp` 方式比较大小。
 - 例如 `ab` 和 `AB` 比较大小，采用 `strcmp` 得到 `AB < ab`。
 - 例如 `ab10` 和 `10AB` 比较大小，都包含数字，且数字大小相等，则按 `strcmp` 进行大小比较，由于字符 `1` 小于字符 `a`，则 `10AB < ab10`。
- 所有不含数字的字符串均小于含数字的字符串。
 - 例如 `ab` 和 `ab10` 比较大小，`ab10` 包含数字，则 `ab10 < ab`。而采用 `strcmp` 将得到 `ab < ab10`。
 - 字符串中可能包含 `0`，如 `a0`。由于 `a0` 包含数字 `0`，根据规则，所有不含数字的字符串均小于含数字的字符串，于是 `a0 > zzzz`。尽管 `a < z`。
- 每个字符串的长度范围为 $1 \sim 30$ ，而其中包含的数字个数范围为 $0 \sim 8$ ，且数字在一个字符串中是连续的（例如 `abc123` 和 `123abc` 都是可能的，而 `ab12cd34` 是不可能出现在输入数据中的）。

写一个程序，用上述规则对一组字符串按升序排序。

输入数据格式及样例

若干个由空格分隔的字符串，每个字符串的长度不超过 30。

```
1 | abc# abc1 abc10 abcd abc2 abc
```

输出数据格式及样例

排序后的若干个字符串，每两个字符串之间用一个空格分开。

```
1 | abc abc# abcd abc1 abc2 abc10
```

备注

字符串的长度要记得包括 `\0`。

调用 `scanf` 函数时，需要判断输入已经全部处理完成，可以使用 `EOF`。

在 Windows 环境中运行程序时，同时按下 `Control + Z` 后，输入回车，表示输入流的结束。