

10352 Count the eWords

The Malfunction Checkers Ltd (MCL) is famous for its extraordinary ability to find faults of a malfunctioning device. Many companies depend on it to find out faults of their newly developed products such as TV, VCR, Radio etc.

Few days ago, the administrator of MCL has come to know that someone has been trying hard to damage MCL's reputation by installing an email virus in the main server of MCL. To track this attempt, the administrator has planned to read all incoming emails. But the employees of MCL are very cautious about their privacy and are opposing the plan. As a result, administrator has to change his plan: instead of reading emails, he is planning write a program to count the words in each email and analyze the result.

What the administrator does not know is that the main server's *C* compiler is already infected. A virus has tactfully damaged some libraries of the compiler. As a result, the `strcmp()` function of the compiler works in an unexpected way. The problems are:

- It ignores the letters in the 3rd position during comparison.
- The function decides its return value considering only the starting 5 letters.

You are to find out the output of the program written by the administrator.

Input

Input contains several emails, each ends with the word '#' which should not be counted. The input file is terminated by end of file (EOF).

Each email contains a numbers of lines of words. The words may contains any "readable" character ("except #") and have maximum length of 20. The number of distinct words is limited to 4100, but any word may appear more than once.

Output

Each email's output starts with 'Set #n:' line, and ends with an empty line.

The output shows in each line a distinct word truncated after length 5, a space, and then the number of times it appears. The lines are ordered in ascending (ASCII) order of the words. The "ignored" (i.e. the 3rd) letter in the word should show the last "ignored" letter. However, the sorting should be according to the first "ignored" letter. For example if the email contains the line 'tid tim tis' then you should consider the three words as one word and print them as 'tis' (the last word in this group), but while producing the sorted output you should sort them considering them as 'tid' (the first one in this group)

Sample Input

```
This is an longinput string
sort the this line if is has soft line break
#
this is a set of sit
#
timk tidk tisk tia tiy tiz
#
```

Sample Output

Set #1:

This 1
an 1
break 1
has 1
if 1
is 2
line 2
longi 1
soft 2
strin 1
the 1
this 1

Set #2:

a 1
is 1
of 1
set 1
sit 1
this 1

Set #3:

tiz 3
tisk 3