It is often helpful for computer users to see a visual representation of the file structure on their computers. The "explorer" in Microsoft Windows is an example of such a system. Before the days of graphical user interfaces, however, such visual representations were not possible. The best that could be done was to show a static "map" of directories and files, using indentation as a guide to directory contents.

For example:

```
ROOT
|     DIR1
|     File1
|     File2
|     File3
|     DIR2
|     DIR3
|     File1
File1
File2
```

This shows that the root directory contains two files and three subdirectories. The first subdirectory contains 3 files, the second is empty and the third contains one file.

Input

Write a program that reads a series of data sets representing a computer file structure. A data set ends with a line containing a single '`*`', and the end of valid data is denoted by a line containing a single '`#`'.

The data set contains a series of file and directory names. (The root directory is assumed to be the starting point.) The end of a directory is denoted by a '`]`'.

Directory names begin with a lower case '`d`' and file names begin with a lower case '`f`'. File names may or may not have an extension (such as `fmyfile.dat` or `fmyfile`).

File and directory names may not contain spaces.

Output

The contents of any directory should list any subdirectories first, followed by files, if any. All files should be in alphabetical order within each directory.

Note that each data set output is marked by the label '`DATA SET` x`:`', where x denotes the number of the set, beginning at 1.

Note also the blank line between the output data sets. Each level of indentation should show a '`|`' followed by 5 spaces.

Sample Input

```
file1
file2
dir3
dir2
file1
file2
]
]
file4
dir1
]
file3
*
file2
file1
*
#
```

Sample Output

```
DATA SET 1:
ROOT
|     dir3
|     |     dir2
|     |     file1
|     |     file2
|     dir1
file1
file2
file3
file4

DATA SET 2:
ROOT
file1
file2
```