

# Index Generation

Most nonfiction and reference books have an *index* to help readers find references to specific terms or concepts in the text. Here is a sample index.

```
larch, 4, 237, 238, 414
+ Monty Python and, 64, 65, 66
+ planting of, 17
Lenny Kravitz, 50
+ going his way, 53
lumbago, 107
mango
+ Chris Kattan, 380
+ storage of, 87, 90
+ use in Nethack, 500, 501
+ Vitamin C content, 192
```

Each index entry contains a **primary entry** followed by zero or more **secondary entries**, which begin with a '+'. Entries will normally be followed by a list of page references, but a primary entry might not be if at least one secondary entry is present (as is the case with *mango*, above). Primary entries are sorted, and secondary entries following a primary entry are also sorted. Sorting is case-insensitive. Page references for an entry are in ascending order and do not include duplicates. (A duplicate could occur if there are two or more identical entries on the same page.)

Your task is to read a document that has index information embedded within it and produce the index. Documents consist of one or more lines of ASCII text. The page number starts at 1, and the character '&' indicates the start of a new page (which adds 1 to the current page number). Index entries are indicated by a **marker**, which in its most elaborate form has the following syntax:

```
{text%primary$secondary}
```

Here *text* is the text to be indexed, *primary* is an alternative primary entry, and *secondary* is a secondary entry. Both '%primary' and '\$secondary' are optional, but if both are present they must appear in the order given. If *primary* is present then it is used as the primary entry, and if not then *text* is used as the primary entry. If *secondary* is present then the marker adds a page reference for that secondary entry; otherwise it adds a page reference for the primary entry. A single marker cannot add a page reference for both a primary and secondary entry. Here are examples of each of the four possible types of marker, which correspond to four of the entries in the sample index above.

```
... his {lumbago} was acting up, so ...
... {Lenny%Lenny Kravitz} lit up the crowd with his version of ...
... Monty Python often used the {larch$Monty Python and} in ...
... when storing {mangos%mango$storage of}, be sure to ...
```

The input consists of one or more documents, followed by a line containing only '\*' that signals the end of the input. Documents are implicitly numbered starting with 1. Each document consists of one or more lines of text followed by a line containing only '\*'. Each line of text will be at most 79 characters long, not counting end-of-line characters. For document *i*, output the line 'DOCUMENT *i*' followed by the sorted index using the exact output format shown in the examples.

Be sure to read [Notes to Teams](#), which has general formatting guidelines that pertain to all problem input files, including this one. Also note:

- A document will contain at most 100 markers, with at most 20 primary entries.
- A primary entry will have at most 5 secondary entries.
- An entry will have at most 10 unique page references (not including duplicates).
- The character '&' will not appear anywhere within a marker, and will appear at most 500 times within a document.
- The character '\*' is used only to signal the end of a document or the end of the input.
- The characters '{', '}', '%', and '\$' will only be used to define markers, and will not appear in any text or entries.
- A marker may span one or more lines. Every end-of-line within a marker must be converted to a single space.
- A space within a marker (including a converted end-of-line) is normally included in the text/entry, just like any other character. However, any space that immediately follows '{', immediately precedes '}', or is immediately adjacent to '%' or '\$' must be ignored.
- The total length of a marker, measured from the opening '{' to the closing '}', and in which all embedded end-of-lines are converted to spaces, will be at most 79 characters.

**Input:**

Call me Ishmael.

\*

One {fish \$unary}, two {fish\$ binary},&red {fish \$ scarlet}, blue {fish\$ azure}. & By { Dr. Seuss }.

\*

This is a {simple } & & { document} that &{ simply %simple

\$adverb

} & {illustrates %vision} &&&& one {simple-minded% simple} {Judge}'s {vision} for what a {document } might { look % vision} like.

\*

\*\*

**Output:**

DOCUMENT 1

DOCUMENT 2

Dr. Seuss, 3

fish

+ azure, 2

+ binary, 1

+ scarlet, 2

+ unary, 1

DOCUMENT 3

document, 3, 10

Judge, 10

simple, 1, 10

+ adverb, 4

vision, 5, 10